The USENIX Association Newsletter

;login:

January/February 1994 Vol. 19, No. 1

Hyping the Information Highway

After our Association's collective effort to bring UNIX to the forefront of our industry, many of our members now find themselves staring right down the barrel of the greatest technological publicity hype since WindowsNT: The Information Highway.

The Information Highway (TIH) is going to bring our nation a cure for all that ails it. People will finally be able to get the information they need, when they need it, delivered to their lap(top), instantly, perfectly, with no effort on their part, and, in all probability, at a cost of pennies per year.

Kids on CNN's Technology Week (94/01/02) knew better: "Who'd have time to scan 500 channels to find what they wanted to watch?" asked one high school senior. Now, 500 channels won't have all new programming – they will be used for timeshifting television shows and "squirting" movies into peoples' TV sets for viewing at their leisure. We don't have enough programming for our current channels.

I'm still hard-pressed to imagine what we'll put on a 1Gb/sec wire if we don't overwhelm it with voice and video (or pictures). People just don't create data that quickly, unless it's mechanically generated – and that's usually fairly boring data (e.g., seismic data from 4,327 seismic stations every few milliseconds or so).

Why does Joe Research need real-time video connectivity to the supercomputer in Japan? I don't see the cost-benefit there. One of my bosses pointed out that people lower their bandwidth requirements dramatically when the real costs are finally exposed. Hardware is costly.

CNN reported that the American Public is willing to pay as much as $10 more per month to join TIH. High-level executives, on the other hand, are touting this as a trillion dollar opportunity. I perceive a disparity (100x for a one-year run).

TIH has some good points. Let's not set expectations so high that no one can reach them. RK

The closing date for submissions to the next issue of *;login:* is February 15, 1994.

Association News

President's Letter ... 3
A Special Thanks ... 4
1994 Elections for Board of Directors ... 5
USENIX Member Benefits ... 5
LISA '93 Conference Reports ... 6
Submissions to USENIX ... 16
Staff Changes ... 16
Vendor Exhibitions ... 17
Community News ... 17
Call for Tutorial Proposals ... 18

SAGE News

SAGE Elections ... 19
SAGE Views ... 20
Elizabeth Zwicky
Factoids: Fun Things to Tell and Know ... 21
Pat Wilson
SAGE Working Groups ... 22
Pat Parseghian
Perl Practicum ... 25
Hal Pomeranz

Features

Response: On Certification ... 27
Bill Hunter, Tony Sanders
So You Want to Hire A Cheap System Administrator 28
Rob Kolstad
Review of BSD/386 ... 29
Lou Katz
What's New ... 30
;login: 50 and 100 Years Ago ... 31
Barry Shein
Imake Rhymes with *Mistake* ... 32
Jeffrey S. Haemer
Imake Response ... 34
Paul Dubois
The Fifth International Olympiad in Informatics ... 36
Donald T. Piele
Computers Could Be Like Autos ... 45
Dave Taber
An Update on UNIX-related Standards Activities ... 39
Nicholas Stoughton

Book Reviews

The Bookworm ... 46
Peter H. Salus
Learning the Korn Shell ... 48
Adam S. Moskowitz
Learning Perl ... 49
Rob Kolstad
Announcements/Call for Upcoming Events ... 50
Publications Order Forms ... 63
Local User Groups ... 72
Calendar of Events ... 75

The UNIX and Advanced Computing Systems Professional & Technical Association

General Information

;login: is the official newsletter of the USENIX Association.

;login: (ISSN 1044-6397) Volume 19, Number 1 (January/February 1994) is published bimonthly by the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710. $24 of each member's annual dues is for an annual subscription to *;login:*. Subscriptions for nonmembers are $50 per year. Second-class postage paid at Berkeley, CA and additional offices. POSTMASTER: Send address changes to *;login:*, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

Contributions Solicited

You are encouraged to contribute articles, book reviews, and announcements to *;login:*. Send them via email to <*login@usenix.org*> or through the postal system to the Association office. Send SAGE material to <*bigmac@erg.sri.com*>. The Association reserves the right to edit submitted material. Any reproduction of this newsletter in its entirety or in part requires the permission of the Association and the author(s).

Editorial Staff

Rob Kolstad, Editor <*kolstad@usenix.org*>
Ellie Young, Staff Editor <*ellie@usenix.org*>
Carolyn S. Carr, Managing Editor & Typesetter <*carolyn@usenix.org*>
Nick Stoughton, Standards Report Editor <*nick@usenix.org*>
Bryan McDonald, SAGE Editor <*bigmac@erg.sri.com*>
Peter H. Salus, Copy Editor
Alain Hénon, Typesetter
Lilia Scott, Proofreader

Membership and Publications

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
Telephone: 510/528-8649
FAX: 510/548-5738
Email: <*office@usenix.org*>

Copyright © 1994 USENIX® Association. The Association acknowledges all trade references made herein.

;login: is produced with Framemaker 3.1 software provided by Frame Technology and displayed on an NCD X Terminal, donated by Network Computing Devices. The system is served by a Sun SPARCsystem 10 server. Output is sent to a 600 dpi QMS 860 laser printer, donated by Quality Micro Systems. The newsletter is printed on recycled paper.

Conferences & Symposia

Judith F. DesHarnais, Conference Coordinator
USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA 92630
Telephone: 714/588-8649
FAX: 714/588-9706
Email: <*conference@usenix.org*>

Automatic Information Server

To receive information electronically about upcoming USENIX symposia & conferences, finger <*info@usenix.org*> and you will be directed to the catalog which outlines all available information about USENIX services.

Tutorials

Daniel V. Klein, Tutorial Coordinator
Telephone: 412/421-2332
Email: <*dvk@usenix.org*>

USENIX Supporting Members

ANDATACO
ASANTÉ Technologies, Inc.
Frame Technology Corporation
Matsushita Electrical Industrial Co., Ltd.
Network Computing Devices, Inc.
OTA Limited Partnership
QMS, Inc.
Sybase, Inc.
UUNET Technologies, Inc.

SAGE Supporting Member

Enterprise Systems Management Corporation

USENIX Association Board of Directors

If a member wishes to communicate directly with the USENIX Board of Directors, he or she may do so by writing to <*board@usenix.org*>.

President:
Stephen C. Johnson <*scj@usenix.org*>
Vice President:
Michael D. O'Dell <*mo@usenix.org*>
Secretary:
Evi Nemeth <*evi@usenix.org*>
Treasurer:
Rick Adams <*rick@usenix.org*>

Directors:
Eric Allman <*eric@usenix.org*>
Tom Christiansen <*tchrist@usenix.org*>
Lori Grob <*grob@usenix.org*>
Barry Shein <*bzs@usenix.org*>

USENIX Association

Executive Director
Ellie Young <*ellie@usenix.org*>

President's Letter — A Tale of Two Companies

by Steve Johnson
<scj@usenix.org>

One of the pleasures of consulting comes from interacting with many different companies. It continually amazes me how different these companies actually are; some manager's jobs seem to involve mostly asking other people for permission, while others seem to have a completely free reign. Some companies make it easy to hire consultants, others hard. Some very rich companies make their highly paid staff get by with outdated and unsupported equipment, while other companies overcapitalize everything.

I saw a very sharp contrast this year, as I had occasion to work with a couple of the larger players in the computer business. These two companies, call them company A and company B, are not direct competitors in most markets, but both produce a wide range of equipment, have annual sales in the multi-billions, and can point to some impressive successes in their history.

My experience with these two companies could not have been more different. I got a call from company A on a Monday, visited them on a Wednesday, and found all the paperwork waiting for me to sign — I actually worked several hours that day. My invoices, in spite of being sent to an out-of-state office, are paid promptly, typically within ten days.

I visited company B about a half-dozen times, each time talking to various different managers. I prepared several different proposals, and finally got a very short-term proposal accepted "in principle." The contract negotiations went on for almost two months, where all the managers agreed to the contract but the corporate lawyers kept finding reasons to send it back. By the time I had signed the contract, I had completed the work, and submitted my invoice. I then started negotiating for the follow-up contract.

Meanwhile, the invoice, also bouncing around out of state, gave rise to some of the strangest paperwork I have ever seen. I had to send back a form agreeing that I would not emit CFC's or other gases harmful to the ozone layer while doing the work. I had to resubmit the invoice with the account number on it, even though I couldn't get an account number without submitting an invoice. I told at least three different people my social security number. Finally, I got paid.

By this time, several months had past and I had negotiated another contract for another small piece of work, preparing several presentations on my first project and my plans for the follow-up. Then the company had a reorganization! My contact spent several months trying to figure out who was responsible for what — when the shoes stopped falling, it was clear that he had lost most of the responsibility for the area in which we had been working. He introduced me to the person now in charge of the area, who was overworked, didn't understand the issues, and clearly didn't want anything to do with me. Meanwhile, the second invoice had gotten lost, sidetracked, the purchase order had been closed by mistake and had to be reopened, and at the time of writing I still haven't been paid for the work I did in September.

As I think back on these experiences, I think there are two major differences between these companies: empowerment and information processing. In many different ways, company A makes their employees feel that the company wants them to get things done and the support staff is there to make things happen. Company B puts an emphasis on going through channels. In company A, problems are identified, brought into focus, attacked, and solved. In company B, problems are passed from person to person, given to task forces, lost in paperwork. Discussion of problems makes people uncomfortable. Company A is in touch with its customers; company B thinks its customers are just like they are (a possibly fatal mistake for the project I was involved in).

In company A, I am working on a project that will probably span five or more years from its conception until the products are fully deployed. I think this is exactly the kind of thing a large, successful company should be doing – good work too large for startups. Company B had at least three significant changes of direction in calendar year 1993 in the project I worked on.

Is it any surprise that company A saw the recession coming, cut way back on hiring, and had very few layoffs in the last several years, while company B has laid off thousands?

The most successful computer companies have all screwed up, and in some cases screwed up very badly, but the ones that survived and grew were the companies that recognized that there

were problems, faced them, and fixed them. The ones that died or are on the ropes developed cultures where problems were ignored, minimized, or "overshadowed by our strengths." One employee in company B, attempting to focus on a product problem, was asked "aren't you a team player?"

Do you work for company A or company B? Of course, company A has some bad departments, and company B some good ones, but, like eddies in a river, eventually these groups go with the flow. As thousands of former Wang, Prime, IBM, etc. employees now realize, it is better to be in a watertight rowboat than in a stateroom on a sinking ship.

If you are in a company B situation, your choice may be as simple as leaving now or being thrown out later. What you can do is to strengthen your resume — try to work in those areas within company B that are most strategic to the industry as a whole (I recently talked to a guy at DEC who had been maintaining their Bliss compilers. He left...) Try to carve out individual pieces of the project, so you can write things like "I built" or "I designed" on your resume rather than "was one of a 25 person team that built..."

Realize that working for a company B can mess up your head, making you feel powerless and unable to do anything significant without the help of many others. Get something in your life where this isn't true – help a family member start a company, get involved in a charity, religious or volunteer organization, perfect a personal or athletic skill, learn to cook, take up painting, learn to play an instrument, or volunteer some time to USENIX. Counter the distorted information flow within company B by seeking direct contacts with customers and suppliers of the company, subscribing to trade publications and reading them, attending conferences (yes, USENIX again). And good luck in your next job...

A Special Thanks

by Ellie Young
<ellie@usenix.org>

As the new year rolls in and one reflects on the past year's activities, it seems like an ideal time to convey our gratitude to the volunteers who gratuitously lent their expertise and support for our conferences, publications, and member services this past year. While there are many volunteers that serve on program committees, coordinate the various activities at the conferences, work in various committees, and contribute to this newsletter and to *Computing Systems*, I would like to make special mention of the following individuals, who have long supported USENIX and made significant contributions in 1993:

The program chairs for our 1993 events, as follows: Rob Kolstad & Dan Geer (Winter '93 General Conference, San Diego); Dave Black (Mach Symposium), David Rosenthal (Summer '93 General Conference, Cincinnati); Dan Geer & Clem Cole (Mobile & Location Independent Computing Symposium; Lori Grob (Microkernels & Other Kernel Architectures Symposium); David Cohn & Peter Reiher (Symposium on Distributed & Multiprocessor Systems IV); Bill Cheswick (UNIX Security Symposium); and Bjorn Satdeva (LISA VII Conference).

Tom Cargill who recently stepped down as co-coordinator for the invited talks program at the 1992 and 1993 general conferences.

At our two general conferences: Bob Gray and Brent Welch who continue to put together the invited talks programs; Ed Gould for coordinating "The Guru is In" sessions; Peg Schafer for whipping the Works in Progress sessions together; and Gretchen Phillips, our terminal room coordinator.

Bryan McDonald who continues to devote many hours to serving as the SAGE editor for this newsletter as well as coordinating efforts for SAGE's budding publications program.

Debbie Scherrer, Jaap Akkerhuis, Neil Groundwater, Pat Parseghian, and Margo Seltzer who served as the nominating committee for the Association's 1994 Board of Directors election.

Peter Salus, Elizabeth Zwicky, Hal Pomeranz, Steve Johnson and Barry Shein for their regular contributions to this newsletter.

And last but not least, the members of the USENIX and SAGE Boards of Directors who spend many hours in providing leadership and governance. And lest we forget, our thanks to their employers, spouses, "so's", and families for allowing them to do so.

USENIX is grateful.

1994 Elections for Board of Directors

by Ellie Young
<ellie@usenix.org>

The biennial elections of the Association will be held in March of 1994. Ballots will be sent to all paid-up members as of February 23, on or about February 28. Members will have until March 31 to return their ballots, in the envelopes provided, to the Association office. The results of the election will be announced in *comp.org.usenix*, at the Summer '94 General Conference in Boston, and in the July/August issue of this newsletter.

The Board is made up of eight directors, four of whom are "at large." The others are the President, Vice President, Secretary, and Treasurer. The balloting is preferential, with those candidates with the largest number of votes being elected. Newly elected directors will take office immediately following the conclusion of the Annual Meeting of the Association which is held in June at the Boston Conference.

As of this writing (Jan. 6, 1994), no nominations from the membership (which are open until January 28, 1994) have been received. The following candidates for potential election to the USENIX Association board of Directors were put forward by the USENIX Nominating Committee (see pg. 3 of the previous issue of this newsletter for their full report):

Board members at large (4 slots available):

President: Steve Johnson, *Melismatic Software*
Vice President: Eric Allman, *University of California, Berkeley*
Secretary: Lori Grob, *Chorus systémes*
Treasurer: Rick Adams, *UUNET Technologies*

Board members at large (4 slots available):

Matt Bishop, *University of California, Davis*
Peter Collinson, *Hillside Systems*
Daniel E. Geer, *OpenVision Technologies*
Andrew Hume, *AT&T Bell Laboratories*
Sue LoVerso, *Thinking Machines Corporation*
Greg Rose, *RoSecure Software Pty Ltd.*
Henry Spencer, *University of Toronto*

We urge you to vote. If you would like to check the status of your membership, please contact *<office@usenix.org>*.

USENIX Member Benefits

As a member of the USENIX Association, you receive the following benefits:

- Free subscription to *;login:* — technical features, system administration tips and techniques, international calendar of events, SAGE News, media reviews, Snitch Reports from the USENIX representative and others on various ANSI, IEEE, and ISO standards efforts, and much more.
- Free subscription to *Computing Systems*, refereed technical quarterly published with The MIT Press.
- Discounts on registration fees for the large, multi-topic Winter and Summer technical conferences, LISA, the System Administration conference, the C++ conference, and the various single-topic symposia addressing topics such as UNIX Applications Development, Security, Operating Systems, High-Speed Networking, and Mobile Computing – as many as ten technical meetings every year.
- Discounts on proceedings from USENIX conferences and symposia and other technical publications.
- Discounts on the USENIX Association book series published by The MIT Press. Now available: *The Evolution of C++: Language Design in the Marketplace of Ideas*, edited by Jim Waldo of Sun Microsystems Laboratories.
- Savings (10-20%) on selected publications from McGraw-Hill, The MIT Press, Prentice Hall, John Wiley & Sons, O'Reilly and Associates, and UniForum.
- Right to join SAGE.

Contact the USENIX Association office at 510/528-8649 or via email *<office@usenix.org>* for help placing publications orders.

LISA '93 Conference Reports

Keynote

by Rik Farrow
<crow!rik@uunet.UU.NET>

John Black started out by describing the tremendous strides taken by computer technology in the last 12 years in price performance, on the desktop a factor of 10, at the server level, a factor of 50, and mainframes vs. massive parallel processors, a ratio of 3,472!

He went on to say that Oracle has eight MPP sites today. MPP means scaling from hundreds of users to thousands of users, from gigabytes to terabytes or petabytes of disk/data, and from 100 transaction/second to tens of thousands. He claims that 14 large companies, including NCube Intel, and Teradata, are preparing large MPP machines for market. Then he asked the attendees if they were ready for this. His contention was that we were not.

Black made the following points about UNIX system administration:

- Most solutions are home grown
- Few commercial products, and those are hard to use
- Policy and procedures are almost always homegrown
- Usually not well documented
- Often inconsistent
- Different in every shop

He claims that proprietary and open systems shops cost about the same amount, using the following breakdown:

	UNIX	VM, VMS, MVS
Hardware/Software	30%	70%
SysAdm/Prog staff	70%	30%

Of course, the audience disagreed with this assumption. Black went on to say that UNIX systems cost 2.2 times as much as proprietary to administer (based on lack of standard tools, policies, and procedures). He mentioned a group of companies which formed the Moses project where they reduced this ratio to 1.64 by introducing standards procedures. Black believes it is possible to run UNIX at 0.3 of proprietary, and if it happened, UNIX would "own the world."

Black then pointed out what he called the "new hire problem." Mainframe shops all use common tools, and have common policies. The learning curve for a new MIS hire is about 15%. But for UNIX shops, it's more like 200% (unlearn 100% and learn 100% new procedures). While most of the attendees felt this was an exaggeration (there is lots you know that is still relevant), he made his point. Only the least common denominator can be carried from one UNIX shop to another.

UNIX is the most productive environment there is. But there are still problems. In MVS, there is one way to do something, so the staff programmers get right to work. In UNIX, there are fourteen ways to do anything, so the UNIX programmers argue about the best way to do it. In the end, the MVS group might finish first.

System administration is getting efficient more slowly than computer power is growing. Sysadmin is a brake on technical progress. So "Don't let John down," said John Black. Running UNIX shops can be cheaper (0.3x proprietary).

After Black opened the floor for questions, most people (myself included) pointed out that the tools that existed today lacked the flexibility to handle the computing needs of UNIX systems. Change takes place at a glacial pace in large, proprietary shops where they all do the same thing. UNIX grew up in a small shop environment, where each system was doing something different. This makes for a much more complicated environment – and makes system administration more difficult.

Vis à Vis Session: Collaborative Network Communications: Using MUDs as Systems Tools

by JR Oldroyd
<jr@opal.com>

Remy kicked off the first session with a very well delivered talk on the use of MUD (yes, that's right – Multi User Dungeons) software as a systems communication tool.

At Northeastern, Remy supports over 1000 users on 300 systems. Many of these users are naive users, but many also are of the budding expert type: computer systems students. He has five administrators who support the network and provide help to anyone who needs it. A big prob-

lem that Remy had to solve was how to provide a reliable and efficient communications tool both between the administrators themselves ("is anyone working on that printer failure report?"), and also between users and the admin group – particularly a problem since the administrators are always on the move (around the lab, or even at home) and so are very hard to track down in person.

The solution was to build a MUD environment. MUD enables a virtual world to be created (containing things like forests and mountains and all sorts of transportable objects), and it provides a means for people to enter the world and interact both with each other (you can talk to each other) and with objects in the world ("Jim takes the printer and slams it against the wall"). All this happens in real-time by means of scrolling chatter on the screen.

With the MUD system, Remy and his admins have been able to keep in easy communication, even when not in their own offices. Even while working in some other location, each admin will log on to the nearest workstation and fire up the MUD. Questions from other admins such as "does anyone know who changed the DNS records, and why?" can get immediate responses and allow everyone to get all jobs done quickly. The tool has proven a major benefit to the efficiency of the group.

There are some downsides, though. The chatty nature of the tool, easily leads to distractions. It takes some discipline to iconify the MUD tool when you really have to get some work done that needs concentration. The text-only nature of the tool can be restrictive. Future work will add multi-media support, including sound and workstation voice channels. As you join other administrators in the virtual world, you'll be able to talk directly with them using the mike and speaker of your workstation.

On the plus side, after a long day of administering, it is quite therapeutic to enter the virtual environment and hang out with the folks.

Perl 5: Tom Christiansen

by Steve Harris

<etnibsd!vsh@uunet.uu.net>

Tom provided an overview of Perl 5, discussing its current status, new features, library and debugger enhancements, and various wishlist items.

Tom's overheads are available (in three sizes) for anonymous ftp from convex.com in */pub/Perl/info/Perl5.[149]up.ps*.

Principal features of Perl 5:

- The interpreter has been almost totally rewritten.
- A few minor incompatibilities between Perl 5 and Perl 4.
- Lexically scoped variables using the "my" keyword.
- References, which are like pointers, but typed– facilitate creation and manipulation of arbitrary data structures, including nested lists, tables, lists of tables, etc.
- Perl with classes: An object is simply a package (e.g., MYOB) containing functions (methods) for manipulating the object's data. An instance of an object is just a reference.

Ongoing stuff: The remainder of the talk focused on ongoing projects that will be done as time permits (get Tom's slides for details).

- GUI Perl – Lots of people want it, neither Tom nor Larry is an X hacker, unless somebody vol-unteers to do it, it probably won't make it in release 5.0.
- Development tools and debugger enhancements.
- New libraries (and general library cleanup).
- Documentation.

Panel: Things that Don't Scale

by John P. Rouillard

<rouilj@terminus.cs.umb.edu>

The panel on "Things That Don't Scale Well" was hosted by Helen Harrison from SAS Institute. Also on the panel were:

Alan Addis – Chemical Abstract Service
Kim Carney – MIT
Steve Hanson – FERMILAB
Evi Nemeth – University of Colorado

This was primarily a question and answer session, however Helen started off with the top 8 myths of large site operation:

1) There exists a point and time where all of the machines on your network are up and running.

2) A user depends solely on the workstation in front of him/her.

3) Today's backups will finish before tomorrows are due to start.

4) All machines can share a central resource.

5) You know all of your users and can communicate with them immediately.

6) You know where all of your machines are and can touch them.

7) Your vendors will provide you with useful tools.

8) You can make all of your users happy.

Some one on the floor suggested a ninth rule:

9) You know all of the sysadmins at your site and trust them.

One major theme was the pros and cons of using the network information service (formally Yellow Pages), as well as a discussion about using hesiod to replace YP/NIS. Backup tape subsystems, energy saving maneuvers, and nfs traffic monitoring/tuning were also discussed.

Source Control

by John P. Rouillard
<rouilj@terminus.cs.umb.edu>

Water Wong presented a modification to the CMU Depot scheme that enables local workstation configuration of the tools available from the CMU Depot. This mechanism provides the best features of the 'Depot,' a single centralized and specialized source for maintaining up to date working software, and adds the ability for the workstation manager to configure the software for the local host/environment.

This also allows people to keep local copies of various collections (software packages), providing cheap mobile computing by allowing the workstation owner to keep local copies of the most-used software for those times when a network connection is unavailable.

Bjorn Satdeva presented some powerful features that make the BSD make utility his favorite for heterogeneous build management. One advantage to the BSD make, is that it knows the type of host that it is compiling for. This can be used to conditionalize the makefile to provide different values for the makefile variables such as CC, CFLAGS, etc. One other major advantage is that the BSD make puts all of the objects for the build outside of the source tree. Thus the source tree can be mounted read only preventing inadvertent modification of the sources. Since the object area does not have to be a resource shared across platforms, parallel builds across multiple platforms can be easily done. This greatly reduces the amount of time that is spent rebuilding software to test fixes.

Jay Ashford with the AIX development team in Austin Texas was a last minute substitute for Barry Archer in discussing the P1003.7.2 Posix draft standard for Software Administration. The first draft document that the standards committee used was provided by Hewlett Packard. The standard that is currently before the committee is divided into a number of sections dealing with: package layout, management tasks, command structure, and management information. The current draft is available in postscript format via anonymous ftp in a subdirectory under: *dcdmjw.fnal.gov:posix/dot7.2*. This document will be available until the voting date sometime in May. Only IEEE members may be involved in the voting process.

ISDN:JR Oldroyd

<jr@opal.com>

ISDN: Internet to the Home

by Stan Kluz

Stan gave a full-session presentation. He concentrated mainly on an overview of what ISDN is, and what the current state of development and deployment is. Towards the end, Stan included some examples of how LLNL is using ISDN to hook up staff working from home.

The majority of Stan's talk was a fairly general introduction to ISDN, what it is and how you can get it. Most of the what it is material can be found in any introductory text book on ISDN. As a rough idea of what was covered, here's a quick overview of Stan's main points.

ISDN has been with us for about 7 years now. It is the subscriber end of the digital communications system that also includes digital trucks and digital switches. Complete implementation of ISDN will take time as switches need to be upgraded (compare this to the time it took to build the railroads, or install electric wiring to everyone). There are two forms of ISDN service: Basic Rate Interface (BRI) which consists of two 64 Kbps channels ('B-channels'), and one 16 Kbps signaling channel (the 'D-channel'). This is your basic digital service. Then there's Primary Rate Interface (PRI) which consists of 23 B-channels and one D-channel. If you purchase ISDN service from your telco, you'll get a wall socket known as the U-interface, where the line comes into the

building. To this you attach an network terminator (NT) from which you can run up to 3,000 ft of lines around the building. Wherever you want a phone or a computer port, you then place a terminal adapter (TA) which provides you with a phone socket or Ethernet or RS232 port as needed. Any phones you attach have to be special ISDN phones, but you can use any Ethernet/RS232 device.

Availability was raised as an issue ("I called my telco, and the sales person didn't even know what ISDN was"). Stan put up some figures with the latest deployment figures for a number of US telcos, and included a bunch of names and numbers for you to call to get the ISDN folk there. Stan mentioned that if you're interested in ISDN, call them and tell them this: the more people who let them know, the sooner they'll do it in your area. Stan showed some figures showing many telcos levels of availability some were currently at 85-95%, most were at 45-60% with 80-90% anticipated for 1995.

Cost was raised. Where Stan Lives (Livermore, CA), the BRI tariffs are $29.50 per month, flat rate, intra-centrex (within the same switch). Measured single line business service was $26.50 per month, plus measured units.

As he concluded, Stan showed LLNL deployment information. LLNL has about 50 staff working at home offices, all with BRI service at home. The combined two B-channels get them 128 Kbps bandwidth, which with suitable compression can get you the equivalent of 400 Kbps throughput. Not and for under $30 per month.

Getting to @

by JR Oldroyd
<jr@opal.com>

Smoot Carl-Mitchell gave a presentation entitled "Getting to @ – Connecting to the Internet." Regrettably, this presentation was pitched at folk who were totally unfamiliar with the Internet, and the presentation included a great deal of material about what the Net is, and what services it provides ("you can do telnet, ftp, email, finger, gopher, Web, and so on..."). Smoot did include a number of interesting graphs showing the growth of the Net over the last 20 years, and user demographics.

The most technical aspect of the talk included comments on how to obtain an IP number, a domain registration, and the difference in Class A, B, and C addresses.

The Myers-Briggs Type Indicator: An Interpersonal Tool for System Administrators

by Steve Harris
<etnibsd!vsh@uunet.UU.NET>

Betty Jacob and Nancy Shoemaker presented a lively and thought-provoking introduction to the Myers-Briggs Type Indicator (MBTI). Betty conducts MBTI training seminars; Nancy Shoemaker has worked as a sysadmin.

Whereas the paper focused on how the MBTI might be used in a Systems Administration context (and some observations on how it should *not* be used), the presentation allowed Betty and Nancy to give us a feel for Myers-Briggs in practice.

Myers-Briggs consists of four measures of *preferences*, which comprise, for example, someone with an extrovert style will do well in a brainstorming session, while an introvert may need time to digest information before coming up with comments and ideas. Both styles are valuable, and, ideally, will work to complement each other.

A few more exercises, questions and answers, and an opportunity to take the MBTI test followed.

A number of questions were raised about the usefulness of Myers-Briggs. Nancy related some of her experiences, in particular, working with a boss with a significantly different style. Betty noted that these tools are hardly perfect, and that it is easy to "cheat," i.e., to produce a score you think somebody wants. Furthermore, using the test to achieve some preconceived group profile is probably misguided at best.

Panel: Help Me! – Questions & Answers

by Steve Harris
<etnibsd!vsh@uunet.uu.net>

Questions were fielded by Tina Darmohray, Rob Kolstad, Jeff Polk, and Elizabeth Zwicky.

Backups

- Exabytes are great.
- AMPEX makes high-end ($$$) unit.
- Sun's "Backup Copilot" uses an undocumented ioctl to turn off synchronous writes during restore. Same functionality is available in *fastfs*, a program posted to the net (try `archie`). But beware, if your systems crashes while synchronous writes are disabled, you might have to newfs and restore from your level 0 dump.

Password Security

- Idea of disabling a user's account after three bad login attempts considered dangerous: "Gee, what if I try to login as my boss three times? Or as root?" Bjorn noted it as a denial-of-service attack.
- Use `npasswd`.
- Use challenge-response boxes.
- Software version of challenge-response system: s/key at *thumper.bellcore.com* (path recently posted to *comp.security.misc*).

What to do when the boss is unwilling to spend allocated money (e.g., for a laser printer).

- Get a demo unit in — users will become addicted.
- Submit requisition when boss is on vacation.
- Put expenditure justification in writing.
- Find a way to discuss problem informally.
- Do NOT go over boss's head.

Maintenance Issues: Do you really save money by eliminating maintenance on clients? On servers? What about self-maintenance?

- In doing the cost analysis, be sure to include the engineering time to fix the problem, and the cost of prolonged downtime.
- A large site can save money with self-maintenance, but new hardware should not be maintained internally until there is enough of it installed to warrant training and spare parts inventory.

Heterogeneous Mail: How to share mail among MACs, PCs, and UNIX boxes.

- Eudora (freely available, ask *archie)* was highly recommended for PCs and MACs. Interoperates well with IP-based mail systems.
- popmail (pop client on PC?)
- trumpet (ditto?)
- quickmail was generally disliked.

Duplicating directory hierarchies — how best to do?

- `pax` (posix `tar/cpio` replacement — has pathname length limit problem) Supposedly handles special files.
- `tar` (pathname length limit problem)
- `cpio` (pathname length limit problem). May not handle special files.
- `dump` and `restore`.
- `gnutar` (works around pathname length limit, but don't abort extraction until tape is completely processed).

Managing Very Large Sites

by JR Oldroyd
<jr@opal.com>

This session, chaired by Helen Harrison, included six speakers each highlighting a concern or a solution they have to a problem associated with managing sites of several thousand users.

John Finke, RPI, discussed "Simon," a product that he, and other folks at RPI have developed to manage large numbers of ever-changing accounts for students. Their system supports current accounts, disabled accounts, and expired accounts: and automatically moves accounts from one status to another. In addition, the system provides support back to the Registrar's office, in that it has helped ensure that users pay their fees on time.

Paul Gerwitz, from Eastman Kodak, discovered that without any controls, the facilities at his site expanded into a heterogeneous mess. A re-engineered business plan, together with an Information Architecture, and standards conformance, is helping him get things under control.

Bruce Nelson, also from Eastman Kodak, had a lot to complain about existing internal lack of policies and standards. He reinforced Paul's message, stating that a Standard Site Plan Book was helping him keep things organized.

Stephen Jaworski, of BNR, described an interesting 4-layer model in which they decouple the workstation platform from the "desktop." Users use their workstation to connect to hosts in the computer room and compute servers, with which they do their work. Large file servers, known as home directory servers, support the intermediate hosts. When users change their office locations' it is now no longer necessary for BNR to move someone's workstation.

Helen Harrison, SAS Institute, described how SAS supports distribution of applications, updates, and site-specific stuff to 925 machines. Their process, called "SASification" involves a distribution server, and a means on each client to obtain the latest distributions from that server, at boot time.

Aaron Schtromas, IBM Watson Research Center, described a similar problem and their solution of doing an hourly "drip" update.

Backups

by Peter Van Epp
<vanepp@sfu.ca>

"The Amanda Network Backup Manager" paper describes a backup system for automatically backing up multiple workstations (via the network) to a machine with a tape drive. The Amanda package uses a disk local to the tape machine to cache date coming in from the network and thus allows the tape drive to be kept streaming. There is automatic scheduling and tape label checking to make sure that the correct dump gets to the correct tape.

The session "Backups: An Introduction To The Art" in fact turned out to be a product presentation on a new product from Delta Microsystems called "freezeframe." At present the product runs on SunOS 4.x and Solaris and on the Auspex.

It consists of kernel modifications to the device driver code that writes blocks to disk. The idea is to create a "snapshot" of the disk that remains consistent while backup occurs.

The entire technical background of the "freezeframe" filesystem was presented, but it omitted here for brevity.

Frozen file systems enable safe backups. A separate read system call provided by freezeframe first attempts to read the block from the freezeframe cache, and if it is not there, then reads it from the disk.

The presenter did not appear to be aware of a fly in the ointment when the timing of freezing, file creation, and dumps falls into a particular pattern.

Leaving the Dinosaur: Moving Applications From the Mainframe to UNIX

by Peter Van Epp
<vanepp@sfu.ca>

This session was interesting (and raised more than a few hackles from the UNIX folks present!). A couple messages came across:

- UNIX has won the war, so: be nice to the mainframe folks, they are only trying to survive.
- UNIX isn't there yet in terms of available utilities, and there is a large culture difference between UNIX folks and mainframe folks.

There are some "bet the business" type activities running on UNIX (e.g., the trading floor on Wall Street). There are a few utilities lacking on UNIX at the moment. Tape mangement systems are one, Hierarchical Storage Management systems are another, and job scheduling and report generation and control (and the volume printing that goes along with them) are a third. Probably all these things are coming (CA UniCenter is a start for instance), but they aren't here yet.

A good portion of the presentation was on how to avoid the culture clash between the MIS/Data Center folks and the UNIX folks (and having come to UNIX from a mainframe background, I know exactly what he is talking about). As far as I can tell, many of the people present correctly realized that much of the "freedom to do what you like, how you like" that exists in a non-production UNIX environment can't survive in an environment where a "bet the business" application is running. This appears to be an environment that many present don't want to work in.

Horses & Barn Doors: Evolution of Corporate Guidelines for Internet Use

by Sally Hambridge *<sallyh@ludwig.intel.com>*
and Jeff Sedayao *<sedayao@argus.intel.com>*

As a result of a specific incident, Intel has put in place a set of corporate policies that cover all staff with respect to use of the Internet from Intel sites. Intel found that it needed to be able to protect itself in the event that an individual did or said something on the Net that in some way upset a third party.

Sally pointed out that the policies were found to be necessary, and were found to be needed in writing. Next year, the Corporate Internet Policy will form part of the Employee Handbook. The guidelines in the policy have helped avoid a lot of potential problems, for example with new employees attempting to telnet back into their old accounts at the university (the policy tells them not to bother trying). There are minor things, too: if you want to send netnews, create a .sig file with a disclaimer in it that you are not talking on behalf of Intel.

The key things with the policy were the need to advertise it well: at Intel, they include it in the newsletter, it is FTP-able, it'll be in the corporate booklet, and there's even a video about the Internet with dos and don'ts. This together with the establishment of a supporting bureaucracy ensure that it works. The bureaucrats have the task of dealing with authorization to access the Net (employees don't just get this – they have to ask for it) at which time they are advised of the policy, and the bureaucracy polices things, helping to locate policy violators and handing out appropriate punishments (which are mostly warnings).

Our Users Have Root!

Editor's Note: Name of this report's author is lost. Please let me know so proper credit can be given!

This presentation was by Laura Kirk de Leon, Mike Rodriques, and Brent Thompson. At HP's main research center, Laura works with 30 other administrative staff to support a group of 1000 workstations, with over 1000 users.

The unusual aspect of their environment is that all users have the root password to their workstations. In many cases, the admin group does not have the root password – and in some cases, the admin group doesn't even have an account on a particular workstation.

The philosophy is to allow all users to run their little environment exactly as they please. Each user sets up their system. Each user creates user accounts. Each user installs whatever application or even system upgrades that they want. Each user does their own backups, or not!

To help support them, the systems group does provide all sorts of standard things: standard configuration files, standard versions of applications, latest OS updates, and so on, and all these are downloadable over the net using standardized and simplified procedures. In fact, they also provide a service called "Ninstall Star" that allows users to automatically grab latest copies of dynamic files, each night. All users are free to avail themselves of these services, but are not obliged to do so.

The whole system works very well. Most users do make use of the standard configuration procedures and downloadable files. And they enjoy the freedom of total control of their systems.

Security in this environment is provided in two ways. All users are responsible for security of information on their own workstations and servers, in much the same way as they are responsible for security of the papers on and in their desks. The systems group provides perimeter security, using firewalls and so on. This is similar to building security services.

Time Expenditure Survey

by Steve Wright

Rob Kolstad presented the results of the Fall, 1992 survey of system administrators. The survey tried to try to answer the question "What do you really do?" The poll was conducted at the LISA conference (about one-third of attendees responded) and by e-mail. The information, while perhaps skewed by the fact that respondents were self-selected and also by the question design, is the best collection of this type of information ever assembled.

Some of the facts from the study (from a 47.5 hr. average workweek):

System administrators spend time:

- helping users 34%
- system maintenance 25%
- personal development 11%
- installation + config 10%
- management 5%
- backup + restore 3%

The "average" site:

- administrators 4.6
- number of users 368
- servers 15.8
- dataful workstations 83.0
- dataless workstations 9.6
- diskless workstations 10.2
- X-terms 21.2
- GBytes on servers 143.3
- GBytes on clients 58.4
- PCs 393 - MACs 72.6

There was a mean of 82 users/administrator. As the ratio increases the amount of service decreases.

The reported numbers are recollections and accuracy could have been improved through the use of logging. Building on this information, a future questionnaire is planned for late Winter over the Internet.

LISA VII - SAGE Open BOD Meeting

by JR Oldroyd

<*jr@opal.com*>

About 40-50 people attended the SAGE Open Board meeting. The board were all present: Elizabeth Zwicky, Carol Kubicki, Steve Simmons (President), Peg Schafer, Paul Moriarty, Pat Wilson, and Pat Parseghian. Steve thanked a large number of folks for the efforts in supporting SAGE over the last year, including Paul Evans (Uniforum Coordination), Tina Darmohray (Jobs booklet), Bryan McDonaldMcDonald (Publications), Bjorn Satdeva (LISA VII conference chair) and Brent Chapman (mailing list support).

Schafer gave the treasurer's report; the bottom line is that SAGE is $7,522 in the black for 1993. A large deficit is anticipated for 1994 due to additional administrative support; USENIX has agreed to subsidize this deficit. The discretionary fund was spent on promotional material and the two awards.

Wilson announced the formation of the <*sage-jobs-offered@usenix.org*> mailing list. Subscribe using <*majordomo@usenix.org*>.

McDonald gave a summary of publications work: SAGE now has 5-10 pages in *;login:*. There are regular columns and he is looking for contributors. The *Job Descriptions For System Administrators* booklet is done. The Salary and Demographics survey results will be added to a future edition of the book. The book will be mailed to all SAGE members not at the conference.

SAGE runs an FTP archive: *ftp.sage.usenix.org*. Mark Verber coordinates this. Past conference papers are on line. Wais and WWW servers to come.

There are currently no formal SAGE Local Groups. However, the board is aware of three local LISA groups: BayLISA (San Francisco bay area), BBLISA (Boston area), and GSLISA/$GROUPNAME (New Jersey). Internationally, there is SAGE-UK and SAGE-AU. Zwicky mentioned that she is a board member of SAGE-AU. Zwicky reported that SAGE-AU had their first meeting in July, in Melbourne. Seventy people attended – same size as LISA I. There is some duplication of SAGE work, and some unique work. SAGE-AU's next conference is in Perth. Neil Todd (audience) mentioned that the French UNIX User Group (AFUU) is looking into forming a SIG for Administration. Simmons reported that tentative discussions are underway regarding the formation of a SAGE-Canada group.

Simmons concluded the meeting with a call for ideas for SAGE work, and a call for more active participants of the group.

Auspex BOF Report

by Ruth Milner
<rmilner@aoc.nrao.edu>

A Birds-of-a-Feather session for Auspex users and other interested parties was held at the November '93 USENIX Large Installation Systems Administration conference, Tuesday Nov. 2 from 6-8pm. There were a large number of people attending the BOF (a sign-up list passed around garnered 81 names, and there were people wandering in and out as well). Discussion continued in the hall after we had to relinquish the room. With this large a group we could easily have continued for longer.

Auspex was well represented with several managers and engineers in attendance. The session started by going through a list of questions prepared ahead of time. [*Editor's Note: Space precludes listing them here. Email* <kolstad@bsdi.com> *for the details*].

The bulk of the remaining time was spent on a survey and discussion, initiated by Brad Peterson, of the backup tools and media that customer sites are using, and what, if anything, they would like to see Auspex provide in this area. The discussion was fast and furious, so I could not keep detailed notes. Some of the highlights of this discussion:

- Tools: Several attendees were people considering buying an Auspex, and they emphasized that the lack of a really good backup tool integrated with the system is a serious omission for a high-end server.

A big preference for a "dump"-based interface was expressed. Whatever method Auspex may adopt, it must provide the ability to restore files from the dumps on foreign systems. There was a discussion of a wrapper to "dd" for the equivalent of 0-level dumps, which would speed up dumping while still allowing individual files to be retrieved.

- Media: Some customers either have, or are looking at, media such as DLT (Digital Linear Tape), which is extremely high speed and capacity. This drive works now on an HP5 SCSI port. The EXB-120 jukebox works on the current SP, although some details are still being ironed out. Optical storage was also discussed.

DMIG BOF

by Peter Van Epp
<vanepp@sfu.ca>

The subject of this BOF was the DMI project being developed by a number of vendors. DMI is a kernel API which will enable hierarchical file management semantics without dictating policy (that being left to the programs that use the API's services). DMIG is the DMIG working group.

Essentially, DMI consists of modifications to the kernel file system calls (such as read and write) that will allow a program to "monitor" a file by issuing a call that will cause a daemon process to intercept operations on that file [*Editor's Note: Sure sounds like watchdogs, doesn't it?*]. There is also a new set of file system calls that provide the same services as the intercepted calls that do not trip the intercept.

The basic idea is that when a file is migrated to secondary storage such as an optical Jukebox, one of these traps will be set on that file. A user reading or writing the migrated file will cause the trap

to the monitor program which will in turn give control to the HSM system, which will use the bypass system calls to move the file from the secondary storage system back to disk, and then remove the "monitor" trap, and let the trapped system call complete (with, of course, a time delay) just as if the file had never been migrated.

The API is being written so that any of the current HSM-like systems will work with the API. Of course, Auspex envisions other applications, too. Auspex has written a reference port and installed it into the kernel that runs on the Auspex file server, but doesn't currently distribute the libraries needed to run it (although they are willing to consider requests for them via email). Join the DMIG mailing list by writing to <*dmig-request@epoch.com*>.

BOF: Mailing Lists and Majordomo

by John P. Rouillard
<*rouilj@terminus.cs.umb.edu*>

The Mailing List and Majordomo BOF took the form of a question and answer session. Brent Chapman started the BOF by advertising the list-managers mailing list at *greatcircle.com*. Send mail to <*list-managers-request@greatcircle.com*> for subscription information.

One of the major points that was brought up by some of the mailing list managers there was that local mailing list exploders should NOT return bouncing mail to the list manager. Instead the bouncing mail should be directed to the postmaster at the site running the exploder. One suggested mechanism was to use resend from the Majordomo distribution to handle the rewriting/resetting of the envelope sender. Something similar can be done with the '-f' flag in sendmail. Simply setting up an owner-listname alias is NOT good enough because the owner-listname only works for local host delivery.

Some basic questions dealing with lists were asked and answered.

For those interested in majordomo the next major release will be version 2.0, and will include configuration file support, and a mechanism for adding pre and post processing commands to the majordomo functionality.

Panel: Topics in Networking

by Steve Harris
<*etnibsd!vsh@uunet.uu.net*>

Questions fielded by Hal Pomeranz <*hal@aqm.com*> and Brent Chapman <*brent@greatcircle.com*>.

DNS Nameserver Maintenance

- Use tools to auto-create the DNS data files.
- If you do edit them manually, keep RCS version files!
- Trick to get at internal data – look at the cache on the secondary nameserver – it is a dump of the internal data from the primary.
- Debugging: Use "dig." Also "doc" (in BIND distribution).
- Documentation: O'Reilly book, "Network Tools Catalog" (RFC) "A Network Manager's Reading List" (netnews FAQ)
- Run BIND 4.9 (other/subsequent versions unstable)

SNMP – how useful?

- Consensus that SNMP is over-hyped; its functionality can be achieved with simpler tools, e.g., ping, traceroute, telnet.
- Trouble alert capability can be implemented with simple tools (e.g., script to do periodic pings).
- SNMP is cumbersome to configure and manage, can give wrong results (e.g., false "system up" indication), and can soak up cycles, net bandwidth.
- However, SNMP is useful for generating pretty pictures and nice management reports – it makes the MIS types real happy.

Wiring

- If you use Level 5 twisted pair, make sure everything is Level 5. Pushdown blocks, jacks, cables, etc.
- Get references as part of installer bidding process.
- Contract should include demo of functionality of *every* line.
- A $1500 TP line analyzer is worth it.
- Patch panels evil – pushdown blocks good – patch panel invites fiddling by engineers, managers. BTW, Level 5 jacks are rated for only 100 insertions – who's going to keep count?
- Use dental tool to trace individual cable in bunch. Get "Brady" labeler for cables. Use spraypaint to color-code cable ends.

Misc:

- Use PPP, not SLIP – Brent argued that Trent Hein is not up-to-date when he lectures that SLIP is superior to PPP at modem speeds. [*Editor's Note: Query to Trent Hein finds him standing by his comments to use SLIP when possible.*]
- NFS is not usable at less than ethernet speeds – well, maybe at T1 speeds, with some tuning.
- ATM – at least 3 to 5 years out. Assuming it doesn't die first. o FDDI – still on performance curve growth: spec is 100 Mb/sec, current is ~40 Mb/sec.
- Network monitoring – use "netstat -i" – collision rate average should be < 0.5%, peak < 5%.

Invited Paper: How Do You Teach Systems Administration?

by Steve Harris

Presented by David Jones, University of Central Queensland, Rockhampton, Australia. David described the difficulties he faced in developing a Systems Administration class at a small university in an isolated city on Australia's eastern coast.

David has had to overcome a number of obstacles to implementing the sysadmin course:

- His own UNIX sysadmin expertise.
- Lack of resources at the University.
- "UNIX' is a four letter word" attitude of CS Department.

The course, which was offered as an elective this year, will become a required part of the three-year "System Services Training" curriculum. As a requirement, it will have to be made available as a correspondence course. To overcome this obstacle, David plans to use Linux as the baseline OS for the class (all students are expected to purchase a 486 PC).

Difficulties

David discussed some of the problems encountered in implementing the course. These included:

- Students have no prior knowledge of UNIX.
- Thirteen weeks is too short.
- The resources at the university are woefully inadequate.
- At what point do you give the students root privileges? This has been pushed back to the middle of the semester.

Assessment

David discussed some of the ways he had developed to assess each student's achievement. These included a 2000 word essay, porting software, problem solving, verbal reports, tests, logbooks, and personal interviews.

SAGE Awards

SAGE presented its first outstanding achievement award in the field of system administration at the LISA conference in November. This year's recipients were Rob Kolstad and Max Vasilatos, in honor of their contributions in organizing the early LISA conferences. .

Submissions to USENIX

The Submission of Papers to USENIX Conferences and Publications and "Non-Disclosure Agreements"

by Ellie Young
<ellie@usenix.org>

In a few recent instances, papers submitted for presentation at one of the USENIX technical conferences were accompanied by a so-called "non-disclosure agreement" (NDA) issuing from the author's employer. The NDA essentially required the USENIX Association and its representatives (including volunteer program committees and reviewers) to sign and return a form warranting that the information contained in the paper in question would not be divulged until the date of the actual conference.

The USENIX Association, as a matter of policy, will not accept any submissions to its conferences, workshops, newsletter, journal, or any other publication when accompanied by any such "non-disclosure agreement" forms, and will return all such submissions to the author unread.

The USENIX Association feels strongly that such NDAs impede the free flow of information upon which its charter is based, impose an undue burden on its volunteer program committees and reviewers, and in general only add another unnecessary layer of administrative paperwork to the already complicated process of bringing technical and scientific work to publication.

The USENIX Association wishes to emphasize that, as a matter of established policy, all submissions are held in the strictest confidence by its program committees and reviewers. We also note that such submissions are protected against unauthorized distribution by the U.S. Copyright Act of 1976 (Title 17, U.S. Code, Section 102), wherein it is made clear that an original work is automatically protected by copyright when it is "fixed in any tangible medium of expression, now known or later developed, from which [it] can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."

Staff Changes

At the Association's headquarters in Berkeley, the following changes in staffing occurred recently:

Andrea Galleni, who was a key member of the executive office staff for the past 5 years in handling member services, resigned this past Fall in order to pursue her undergraduate studies full-time. Recent reports indicate that she is thriving at San Francisco State University. Lilia Scott, a recent graduate of Hampshire College in English Literature, has been hired as her replacement.

Cynthia Deno, who handles the promotion of our conferences and vendor displays, will no longer be handling the latter activity (in order to devote more time to her 2 year old). Peter Mui (who until recently was with O'Reilly & Associates) will be handling the exhibits (see page 17).

Vendor Exhibitions – Should you Exhibit?

by Peter Mui
<display@usenix.org>

Vendor Exhibitions are held at selected USENIX conferences to showcase products we feel are of interest to our membership. We try to have unique exhibitors that you might not be able to see at another show because of their specialized appeal or the size of their company. Some kinds of companies we like to have exhibit are:

- Regional companies geographically local to our conference site
- Smaller companies with interesting or innovative products
- Divisions of corporations i.e., technology-driven divisions of large companies with a specific interesting product to display

Many USENIX members may work in environments that fit these criteria. Should you or your company be exhibiting at USENIX Conferences? We try to make it as easy as possible for a small company, or a division of a large company, to exhibit, with none of the hassles associated with going to a traditional trade show and at a fraction of the cost. The exhibitor's fee is scaled to the number of expected attendees, and includes all of the furniture and drapery for tabletop presentation, which is set up in advance.

In 1994, we will be having vendor exhibitions at the following conferences:

Winter Conference, January 17–21
San Francisco, CA (1500 attendees)

C++ Conference, April 11–14
Cambridge, MA (300 attendees)

Summer Conference, June 6–10
Boston, MA (1500 attendees)

LISA Conference, September 19–23
San Diego, CA (1000 attendees)

For more information on exhibiting at these shows, contact Peter Mui at the USENIX office: 510/528-8649 (phone), 510/548-5738(FAX), or send email to *<display@usenix.org>*.

Community News

Marc Mengel *<mengel@fnal.gov>* and Laura Appleton *<appleton@fnal.gov>* are now officially engaged to be married.

Jim Duncan *<jim@math.psu.edu>* of the Duncan Saunders Group announces their latest product, the VAL 1193. After an intense day-long session at Centre Community Hospital on Wednesday, the 17th of November 1993, "Valerie Anne," as she's affectionately known to her admirers, was unveiled to a small but rapt audience at 5:58 PM, EST.

Gretchen Phillips *<phillips@acsu.buffalo.edu>* and John S. Quarterman *<jsq@tic.com>* were married on December 31, 1993.

This Space is Available

Does your company have a product or service that would be of interest to USENIX members? If so, a limited number of pages are now available in this newsletter for advertising. Please contact Diane DeMartini at the USENIX executive office via phone 510/528-8649, FAX 510/548-5738, or email: *<office@usenix.org>* for ad rates and availability.

Call for Tutorial Proposals

by Dan Klein
USENIX Tutorial Coordinator
<dvk@usenix.org>

In an effort to continue to provide the best possible tutorials to its membership, the USENIX Association is soliciting proposals for future new tutorials (a policy statement concerning the USENIX Association is at the end of this post). The

tutorial proposals can cover any subject, ranging from introductory to advanced materials, although one should avoid overly introductory materials (i.e., a one day tutorial on "Introduction to C Programming" is not what we are usually looking for).

Previous conferences have included tutorials on such diverse topics as UNIX Network Programming, X Toolkit Intrinsics, Topics in System Administration, Mach Virtual Memory Internals, System V, Berkeley, and OSF/1 Kernel Internals, Tcl & Tk, and Software Contracts and Intellectual Property, among many others. Tutorial instructors are remunerated for their presentations, and have their registration and reasonable expenses paid for.

Tutorials usually run for a full day (6 hours of class time plus morning, lunch, and afternoon breaks), although we are currently experimenting with half day (3 hour) tutorials. A proposal should include a statement of what you want to teach and a coherent outline to your tutorial (not simply a list of what you want to cover, but the order in which you want to cover it, with an estimate of the amount of time for each subject). Because a tutorial lasts on the order of 6 hours, we need to know that you can comfortably fill that time, but not overfill it (i.e., that you won't suddenly discover at 4:30 that you have another 3 hours of slides left to present).

If you have any supplementary materials to distribute (e.g., copies of papers, shell scripts, source code, illustrations, etc.), give an indication of the volume of supplementary material, and a rough count of the number of slides you will be presenting during class. (Historically, a typical tutorial has between 75-200 slides, along with up to 200 pages of supplementary material). If possible, include a couple of sample slides (one with text, one with a graphic) with your proposal. If you have a complete or draft course already done, a copy of the current materials would be most useful.

We also need to know if you will be presenting or distributing any source code. If so, is it copyrighted by someone other than you? If you do not hold the copyright, you must be able to demonstrate that you have permission to use this material (this may be dealt with by requiring course attendees to have a source license). Because the USENIX tutorials fall outside of the "fair use" clause of the U.S. copyright code, the same rules apply for supplementary papers or reports.

Finally, your proposal should also include a summary of your previous teaching or lecturing experience, as well as a couple of references (that is, one or two people who have seen you teach who we can contact). These may be your students, supervisors, or colleagues.

Remember, this is just a proposal, so nothing you submit will cast in concrete. You may later decide to change some ordering of materials, or we may suggest some changes. You needn't worry about getting it perfect the first time around. What we are trying to do is get a very solid feel for what you are offering. You must sweat out some of the details, but needn't go too crazy over them.

All tutorial proposals are kept in mind when the tutorial program is chosen for a major USENIX conference or for one of our smaller workshops or symposia. If you feel that your proposal would be especially suited for a particular venue, please note that in your cover letter. Please send your proposals to *<dvk@usenix.org>* or by physical mail to:

Daniel Klein
USENIX Tutorial Coordinator
5606 Northumberland
Pittsburgh, PA 15217-1238

Be sure to include an electronic and postal address and a phone number. All proposals will be acknowledged upon receipt.

SAGE Election Results

by Ellie Young
<ellie@usenix.org>

These are the results of the elections for the Board of Directors of SAGE for the 1994-95 term.

There were 342 ballots cast, and these are the results:

Pat Wilson	243	Elected
Elizabeth Zwicky	238	Elected
Paul Evans	233	Elected
Paul Moriarty	198	Elected
J.R. Oldroyd	147	
Hal Pomeranz	137	

The directors elected will serve two year terms covering 1994 and 1995. Three directors (Steve Simmons, Pat Parseghian, and Peg Schafer) return to the SAGE Board of Directors for 1994 to complete their two-year terms.

The SAGE Board of Directors will choose its own officers at the conclusion of its next board meeting at the USENIX Winter '94 Technical Conference in San Francisco, CA. Please check with the USENIX office *<office@usenix.org>*, the SAGE board *<sage-board@usenix.org>*, or the *comp.org.usenix* newsgroup for more information about these meetings.

Show Your Support: Order Your Official SAGE Polo Shirt Now!

SAGE has designed a new organization shirt. The polo shirt is a Land's End (better-built) 100% cotton mesh polo, available in mountain green with a cream SAGE logo.
Shirts are available in the following sizes:

Men's regular:	S: 34-35, M: 38-40, L:42-44, XL: 46-48
Hemmed:	1092-9217
Banded:	0500-2218
Women's regular:	S: 6-8, M: 10-12, L: 14-16, XL 46-48
Hemmed:	1405-8212
Banded:	1405-9218

All shirts are $24.50 each. Order yours now! Not only will you be the proud owner of this beautiful shirt, you will also be promoting SAGE.

Be creative! Any Land's End item (luggage, sweats, caps, etc.) can carry the SAGE logo.

To order the shirts or other items with the SAGE logo, call Land's End Corporate Sales at **800/338-2000**, give them the logo number (#935025) and specify your size.

SAGE Views

System Administration Tools Your Vendor Never Told You About: The Soda Machine

by Elizabeth Zwicky
<zwicky@erg.sri.com>

Health craze or no health craze, most system administrators resort to some sort of fizzy caffeinated beverage at some point. Mythology says that it's usually Classic Coke, and my experience bears that out, but there are the Pepsi holdouts, and people who drink anything from Dr. Pepper to Dr. Brown's Celray. In most workplaces, you can buy some of these, generally for an excruciating amount of money (would you believe SRI wants 75 cents for a Coke?), or you can bring your own and put them in some sort of a refrigerator, where they promptly get stolen.

Get yourself your own soda machine. If your workplace provides one, talk to whoever runs it about what the rules are; generally you can set up a 'private' machine, so you'll need to put it in an access-controlled area. You can buy machines used if you look in the classified ads in your newspaper, but if your shop can live with a vendor monopoly, your local distributor will probably loan you one for free, as long as you buy the stock for it from the distributor. You tell the distributor how much you want to charge, and you get a machine with the price pre-set, but pretty much everything else left where you can rearrange it. As long as you don't damage the machine, play with the changer, or put soda bought from elsewhere into it, the distributor doesn't much care what you do. Since soda machines are designed for modularity and repairability, you may find that the ethernet soda machine is easier to create than you thought.

The distributor will deliver the soda, but it's up to you to put it in the machine. That's part of what makes it cheaper (usually low-volume prices for delivery are about the same as the absolute bottom grocery-store prices, or less than half the standard machine price). This lends itself to new sorts of disasters: I now know what happens if you punch a small hole in the side of a recently shaken soda can, say by dropping it from a height onto a sharp corner. It's the same thing that happens if you blow up a balloon and let go, only messier. Ever seen three programmers chasing an escaping Coke can? On the other hand, I have a fallback job skill now for times of real desperation.

More Metaphors to Live Without: Everybody's VCR is Blinking 12:00

by Elizabeth Zwicky
<zwicky@erg.sri.com>

To give the whole line "Why, today's technology is so hard to understand that everybody's VCR is blinking 12:00." The lesson here is supposed to be that technology is outstripping human ability; it's certainly meant to be a metaphor and not a statement of fact. I can see my VCR from where I write, and it is not blinking 12:00. This is probably because it is showing the tape counter, since "15:88" is not a possible time in my time zone. I strongly suspect that if it were trying to display the time, it wouldn't be the correct time, anyway.

Is this because nobody in my household is capable of setting the time on the VCR? No, it isn't. While I joke about my inability to deal with things that have no command line interface, I can actually struggle through making the VCR work, and my housemate does timed tapings without batting an eyelash. Furthermore, the clock in the guest bedroom was blinking for 3 months, and it was designed by someone who believed that it was extremely important that one be able to set the time. So important that the time is set with hour and minute advance buttons on top, conveniently handy to the snooze bar and the alarm set buttons. Nobody could ever attempt to do anything with this clock without learning, completely inadvertently, everything there is to know about setting the time on it, and it still sat there patiently blinking for months on end.

Given that we are competent to change the time, why do our clocks blink? To start with, there are at least 28 clocks in my house, of which 9 are completely dependent on wall current and reset every time the power goes out. Fortunately, the Bay Area is very short on thunderstorms, but it has the usual number of suicidal squirrels, overenthusiastic heavy equipment operators, and power company employees having really bad days, which lead to a number of blinking clocks. Furthermore, out of all the clocks, only 3 believe in daylight savings time (the rest were obviously designed for use in Arizona).

Changing the time doesn't seem like such a big deal, until you've discovered that you own 28 clocks and there are 25 separate ways of resetting them. (We were clever enough to buy the answering machines as a matched pair, and three of the computers run

the same operating system.) There is no principle which all the designers agree on. The VCR is admittedly a particularly poor example, designed with the idea that what you really need is more buttons, and if the labels have to be scrunched to make space, well, they didn't make that much sense in the first place. It wouldn't make much difference if it was the only clock we had — it's not that difficult. None of them is that difficult to reset; not the clock that takes both hands, not the answering machines which insist in talking to you as you try to reset the time, not the microwave which uses the button that says "minutes" to reset the hours, not the air filter with the knob to set the time with (which only works if you spin it fast enough initially). Taken together, however, they represent at least an hour of time spent playing button, button in order to make certain that the time is right.

Over time, one learns to prioritize one's clock resetting. Nobody really cares what time the clock in the guest bedroom shows, unless we have guests, so we only set it when we're expecting people. The VCR generally shows the tape counter, and when it doesn't, it shows the time in itty-bitty blue numbers that I can't read from any significant distance, so we only reset it when we want to do timed tapings. (If I want to know what time it is in the living room, I turn on the TV.) The clock on the conventional oven runs on a 65 minute hour, so there's no point in setting it at all. (If I want to know what time it is in the kitchen, I look at the coffee maker.)

There may be lessons we can draw from the amazing numbers of VCRs with the wrong time on them; perhaps the most obvious is that most people don't really care what time their VCR thinks it is. A second one is that VCR designers assume that human interfaces don't matter much in convincing people to buy a VCR, presumably correctly, since thousands of blinking 12:00s have not reduced the popularity of the VCR. But assuming that this is a high-technology problem seems bizarre. In fact, the generalization in our household is that the more high-tech complexity a device contains, the easier it is to deal with setting the time on it. The Sun sets it own time, including dealing with daylight savings time (well, not perfectly — but at least it resets the time). The rest of the computers are Macintoshes, which at least keep time without requiring wall current and provide a relatively easy way to set the time. And the peak of high-tech complexity in timekeeping has got to be at work, where we use a radio clock and NTP, and only reset the time when clock chips fail.

Meanwhile, I have no interest in worrying about whether or not people can work their VCRs; I'm busy worrying about whether or not they can adequately work their coffee makers. Forget annoying blinking, coffee makers are capable of spewing boiling water! And judging from my experience at work, VCRs blinking 12:00 will be lost in the mists of history by the time people become consistently able to turn the heat off when the pot is almost empty, and leave it on when the pot is full. In a world where people put a paper towel between the pot and the burner (the bottom of the pot was wet, you see), why are we blaming the problems of modern life on technology?

Factoids: Fun Things to Know and Tell

by Pat Wilson
<paw@jekyll.dartmouth.edu>

These are the very preliminary results from the 1993 LISA Salary Survey.

Statistics from the first 279 surveys entered:

#respondents: female: 62, male: 217
mean of all salaries: $49,826
(female: $47,440, male: $50,966)

Women reported salaries ranging from $15,000 to $75,000, while men's ran from $16,000 (non-US) to $240,000(!).

Women's salaries tended to rise slightly with age, as did men's, but men in this survey still reported slightly more income, on average:

Women:

21-30:	mean: $39,394
31-40:	mean: $49,753
41-50:	mean: $54,281
>50:	mean: $43,000

Men:

11-20:	mean: $21,000
21-30:	mean: $45,096
31-40:	mean: $53,850
41-50:	mean: $57,178
51-60:	mean: $53,333

New York City and the surrounding area still looks like the place to make big bucks (average $133,750 in NYC, and >$70,000 elsewhere), but California isn't too bad either, averaging somewhere in the $60K-$80K range. There's quite a geographic spread on the low end of the spectrum — further analysis is needed even to begin to see the pattern here.

Much more work will be done with this data (including entering the rest of it!), and a more "valid" set of numbers (with pretty charts and graphs) will appear soon in *;login:*. Please do take these numbers with a grain of salt — note that none of the analysis so far takes into account variables such as years of experience, possession of (or by) an advanced degree, academic vs. commercial sector, etc. Other data also collected by this survey will be analyzed and appear in the future.

Feel free to send questions or comments to me: <*pat_wilson@dartmouth.edu*>.

SAGE Working Groups

by Pat Parseghian
<*pep@research.att.com*>

The following text was distributed to all SAGE working group chairs, and all working group members, in late October, 1993.

History of the Working Groups

When SAGE was created in 1992, the founders formed a few "working groups" to study issues that affect system administrators and to make recommendations to the board, and the organization as a whole, on what SAGE could do for the benefit of its members. One example is the sage-jobs group; they have developed a set of job descriptions to help define the skills and responsibilities of system administrators. This work will not only improve hiring expectations, we hope it will help managers understand, and better evaluate, the system administrators already on the job.

In June, 1992 the founders of SAGE met with a larger group of system administrators who were interested in helping the organization to grow. During that meeting, the group identified many new issues that SAGE might pursue and it seemed natural to form additional working groups to study them. The person who raised the issue typically agreed to lead the group.

A few months later, SAGE members and board members began to feel that we had too many working groups: We worried that the organization was spreading itself too thin and losing focus. The working groups were not in regular contact with the board.

What is a Working Group?

SAGE has a board of directors, but those seven people can accomplish just so much. SAGE has many members (more than 1,000 today); it is very difficult for so many individuals to interact and achieve any goal. Working groups fit in between: smaller units that can tackle projects and carry on discussions. A working group is more than a group of people who share the same interests or face the same problems; a working group is a group that wants to do something with those interests or solve those problems.

By joining a working group, an individual SAGE member has the opportunity to influence the direction of the organization (and, we hope, the profession as a whole). In addition to contributing their knowledge, perspectives and skills, members learn from their interactions with the group.

Like so many organizations, SAGE is indebted to the many enthusiastic volunteers who donate their skills and time to work on projects for the benefit of all. We rely on our volunteers, but our volunteers also need to realize that they are accountable to all SAGE members for the work they take on.

Working Groups in Practice

The goals and purposes of some working groups are well-defined: develop job descriptions (sage-jobs), coordinate publications/newsletter (sage-pubs), set up electronic information distribution services for documents and tools (sage-online). Some of the goals of other working groups are not as clear. SAGE has a good picture of what some groups are doing because they keep the board well-informed about their activities. In some cases, SAGE becomes unsure what a group is doing when that group is difficult to contact.

After observing and interacting with the working groups for about a year, the board felt it was time to establish some guidelines for the groups. Four board members formed a subcommittee to draft a charter for the working groups, which they presented to the rest of the board at the June 1993 meeting. Before the meeting, Steve Simmons

(President) independently asked the leaders of the working groups for status reports — not only on their activities, but also on the future of their groups. Only six of the fifteen groups responded. Of those, only two reported ongoing activities. Three of the others acknowledged that they were basically dormant and needed direction, or were perhaps more accurately characterized as discussion groups. The last reporting group proposed disbanding.

After reviewing the charter and the working group responses, the board felt that the charter was "on target." That is, it attempted to remedy many of the problems that the existing working groups reported. The charter was distributed to the working group chairs on July 1, 1993 and published in the October, 1993 issue of *;login:*.

The Working Group Charter

As submitted, the charter was an attempt to describe succinctly a framework for the working groups. The committee tried to keep the document simple and brief, but unfortunately it was not well-received. Most of the objections were to be resolved by clarifying the original proposal, rather than making fundamental changes.

In addition to Working Groups, a different type of group was also proposed: Discussion Groups. Some of the existing working groups are better described as discussion groups: offering a forum for people with similar concerns to share ideas, without a particular goal in sight.

With this background in mind, let's take a look at the charter the committee developed. The committee is eager to refine the charter, if necessary, and to start working within its framework.

The first item, *What is a Working Group?*, pretty much describes the status quo: Each working group has a mail alias. Most of the subscribers listen to the discussions and a few participate actively. Groups need to focus on issues that are important to SAGE members. To succeed, a working group needs more than one active member.

To address the problem of "drift" among working groups, a second item, *Working Group Structure*, describes the attributes that characterized successful working groups. Just as it's hard for the SAGE members to act as one body, it can be hard for a large working group to make progress, too. If a few members tackle a problem instead, they can summarize their ideas, get comments from the larger group, and so forth. This helps the group at large by insulating them from low-level, detailed discussions. It should help the sub-group, too, by enabling them to hash out the low-level details without seeing lots of similar comments or requests from the larger group. It seemed natural to set up a mail alias to support this structure, so sub-group members could communicate easily and to allow the larger group to pass their comments along to the sub-group. By asking the group to set milestones and a schedule, we hoped to encourage them to partition their tasks.

The most successful working groups have leaders who can and do devote a large measure of time and energy to the group's activities. We tried to capture this successful trait in the section called *Working Group Leadership*. When progress trails off, the leader of the group needs to step in and stir things up. Similarly, any given discussion should not drag on indefinitely. The role of a leader is to summarize and help the group achieve a consensus. A working group leader should interact regularly with the SAGE Board, sharing information about the group's activities as well as taking suggestions from the Board back to the group.

The committee was not looking for ways to create extra work for people; no expectation exists for long monthly reports. The problem that was to be solved was the lack of reports from working groups. Of course, everyone is busy and we are all experts at procrastination. The committee's idea was that if we establish a requirement for a monthly report, that report might be "nothing new to report this month," and that's fine. But if there was nothing new to report last month, and nothing new to report next month, the group would be indicating trouble. Having a monthly report is a way to help the leader see whether the group is really on track. Milestones are likely to be set at intervals longer than a month: but if we ask groups only to report at those intervals, we may realize too late that nothing happened. We want to report regularly to the entire membership on working group activities. We hope that out of *N*working groups, one or more will have provided some newsworthy material in the months between publication deadlines. We have no desire to publish "no progress" reports.

The next section is *Working Group Rewards*, or "What's in it for a working group?" SAGE is a volunteer organization, so a successful working group can expect some fame (but not fortune). Working groups determine not only the direction of SAGE but also influence the future of system administration as a profession. The benefits to individual members of a working group are indirect.

Many of the current working groups can be better characterized as discussion groups. We aimed to describe this alternative in the section, *What is a SAGE Discussion Group?* A discussion group is a way for SAGE members with similar interests to find one another and share information. We think this is a valuable benefit for our members. We also hope that the discussions may become focused, that goals for the group may emerge, and a discussion group may turn into a working group.

If a subset of members approaches the board, wanting to form a discussion or working group, it's up to the board whether SAGE will support the group (with a mail alias, etc.). For example, we won't commit SAGE's resources to support activities that are contrary or irrelevant to the purposes of the organization. I think we can rely on the board's discretion here — if not, then the members can elect a new board at their next opportunity. In addition to specifying how a discussion group gets created, we felt it was equally important to note when a discussion group should be pronounced dead.

Short of asking that board members serve as liaisons for the groups, we didn't think of a way to be aware of what's happening in a discussion group (since we don't ask discussion groups for reports). We wanted to make sure that each working group had a close relationship with at least one Board member, so we formalized that in the section *SAGE Board Liaison*.

Working group members are accountable to SAGE, so it made sense to us that only SAGE members should participate in working groups, as described in the section *Group Membership Eligibility*. We also feel there is a place for newcomers, who may not be members, and so we decided not to restrict participation in discussion groups. Non-members who become involved in discussion groups may later decide it's worthwhile to join the organization.

Working groups need to be accountable to the organization; SAGE is relying on each working group to achieve its stated goals. In the section titled *Group Reviews*, we explained that the Board will review the groups on a regular basis and take action to get the group back on track, if necessary.

What's Next?

There was a Working Groups BOF at the LISA conference in November, hosted by Lee Damon, chairperson of the Policies Working Group. Many of the working groups were represented at the BOF, and we spent some time answering questions about the charter before breaking up into individual working group meetings. The BOF reinforced our view that the charter we drafted is a sound one, as we found that it addressed the questions and suggestions that came up during the BOF.

The next step is to assign a Board Liaison for each group and to review the groups. As part of the review, some discussion groups will emerge and we hope the remaining working groups will regain their focus. Each issue of *;login:* will feature some reports from working groups on their activities.

Perl Practicum

You Say 'rsh' and I Say 'remsh'

by Hal Pomeranz
<pomeranz@aqm.com>

Last time I showed how Perl can emulate many of the more common UNIX filters and information gathering tools. While you spend some time "reinventing the wheel," the payback is a much more portable script. At times though, you simply have to invoke an operating system command. This is where you start running into real portability problems. What directory does the command live in and what arguments does it take? In some cases, even the name of the command is different — on some System V machines, rsh is the restricted shell and remsh executes a command on a remote machine. This column is dedicated to helping you navigate the morass of UNIX dialects and end up with a Perl script that will run on most of them. This column is a little short on Perl, so if you are not interested in maintaining scripts across multiple architectures, then it may not be the article for you.

Where Am I?

The first trick is finding out your machine's architecture. Many systems implement a /bin/arch command which gives some sort of unique identifier to indicate what manufacturer and what architecture type your script is running on. The arch command does not give operating system revision information (sometimes useful), however, and is far from universal. Your best bet is to try /bin/uname.

You can get a machine's hostname, OS type, and hardware architecture with the following command:

```
($host, $os, $arch) =
   split(/\s+/, '/bin/uname -nrm');
```

This works on every UNIX machine I have ever used except Convex machines, which for some strange reason simply do not implement uname. For machines with no uname command, you will just have to build up a list of special cases based on /bin/hostname. If you have a lot of special case machines, you could build up a static associative array by hostname:

```
ENV{'PATH'} = "/bin:/usr/bin";
 %machines = ("convex1", "ConvexOS:
   Convex", ...);
($host, $os, $arch) =
   split(/\s+/, '/bin/uname -nrm');
unless ($host) {
   chop($host = 'hostname');
   ($os, $arch) =
         split(/:/, $machines{$host});
   die "Unknown: $host\n" unless ($os);
}
```

Note that I left the call to the hostname command as a relative path (hostname can live in /bin or /usr/bin depending upon the flavor of UNIX you are using, but I have never found it anywhere else). If you are going to use relative paths in your script, make sure you set the $PATH variable in your environment to a list of known "safe" directories or you will be susceptible to Trojan Horse programs. Never have the current directory (".") or a user home directory in $PATH.

The problem now is that the $os and particularly the $arch values are some strange text string that was meaningful to the vendor, but not necessarily all that humanly intuitive. For example, on SGI machines $arch will be something like "IP\d+" while Amdahls return numeric codes like "580." You will just have to survey all your machines to know exactly what values to expect.

Once you have identified your machine type, you can choose appropriate defaults and then modify them per architecture and OS release:

```
$bigwords = 0;
$gooduucp = 1;
$confdir = "/etc";
$ps = "ps -e"

if ($arch =~ /^sun/) {
   $ps = "ps -ax";
   $gooduucp = 0 unless ($os =~ /^4\./);
}
elsif ($arch =~ /^IP\d+$/) {
   $confdir = "/usr/etc";
}
elsif ($arch =~ /^CRAY/) {
   $bigwords = 1;
}
   :
   :
else {
   die "$host: unknown arch $arch\n";
}
```

Suns use the Berkeley style ps command (unless you are running Solaris 2.x — check $os). Older Suns use a brain-damaged UUCP. SGI machines put some of their configuration files in /usr/etc instead of /etc. Crays have big words, so we need to be careful for bit-shifting operations. It is a good idea to trap for unrecognized architectures.

Doing it Once

If you have a large number of Perl scripts, it may become cumbersome to repeat this same conditional over and over again. There are a several ways to approach this problem.

One choice is to implement a "universal" configuration by creating a giant conditional which properly sets defaults for all of your Perl scripts. Place this file in the same location on all of your machines, and your scripts can use the file either with require or

```
eval { do "$configfile"; };
die "Error in $configfile:\n$@" if ($@);
```

Remember that if you use require, the last statement in the file must evaluate to true. Most packages simply put

```
1;
```

as the last line of the file.

If you have many architectures and many Perl scripts, the conditional can become quite large. On the other hand, you only have to maintain a single file, and it is quite straightforward to bring in a new architecture and port all of your scripts in one fell swoop.

A second alternative is to have a configuration file per individual machine located someplace like /etc. You can then use simple assignments rather than having a large conditional. While this may seem like a great deal of effort, chances are you will only have one file per architecture, or perhaps a few per architecture if you have wildly varying OS releases installed. You can distribute the "master" files from a central location to individual machines using something like rdist. You might even consider writing a "meta-configurer" script which would run out of cron and automatically build configuration files for each machine (a similar program for Bourne shell scripts was presented by Bob Arnold at LISA V[1]).

A third approach is really just an amalgam of previous ideas. Place architecture/OS specific information in separate files, but in a single location available to all machines. By naming the files appropriately, it is easy for you scripts to grab the right one:

```
$configdir = "/usr/local/configs";
($host, $os, $arch) =
                        split(/\s+/, '/
bin/uname -nrm');
die "$host: no config file $arch.$os\n"
                        unless (-f "$con-
figdir/$arch.$os"); eval { "do $con-
figdir/$arch.$os"; };
die "$host: config error:\n$@" if ($@);
```

In this case, all config files are located in /usr/local/configs and are named by the strings returned as $arch and $os by the uname command.

Whatever method you choose, you must be extremely careful to avoid name collisions with variables in the scripts which pull in the configuration files. I tend to use lowercase variable names in the scripts and reserve all uppercase variables for configuration information.

Conclusion

This probably seems like a great deal of wasted effort if you are not a system administrator at a large site or only maintain one or two architectures, and you are absolutely right (but I did try to warn you way back in the first paragraph). If you are a large site administrator contending with a wealth of Perl code, these techniques can simplify your life immeasurably.

1. Arnold, Bob, "If You've Seen One UNIX, You've Seen Them All", LISA V Conference Proceedings, 1991.

Response: On Certification

Opinion by Bill Hunter
<bill@Access.COM>

I read the article in *;login:* (November/December 1993) regarding certification, and I tend to agree more with [Kolstad's] point. I personally am Sun Competency 2000 certified (my company is a master reseller/distributor for Sun, so it's required we have some people certified) and I am also a CNE. I received my Novell CNE a few years ago, and all it proves is that I can sit in a class and take a test. I am by no means terribly Novell literate. The Novell program makes (or made) sense because in the world of PC networking, they are "it." They control a huge portion of the installed base.

UNIX workstations on the other hand tend to show up as large heterogeneous networks, and single vendor certification means very little.

Sun does have an edge with regard to market share today, but it doesn't compare with that of Novell over other PC networking software. I wouldn't even know who is Number 2 in PC networking.

I don't think there are enough "accepted standard" procedures in the UNIX world to make a reasonable program for certification. John Black's keynote address at the LISA conference alluded to this, and until something like that is in place, there will be no way for employers to gauge someone's capabilities with regard to system administration.

One company, Computer Associates, is moving their mainframe tools to the UNIX desktop and actually shipping them with some HP 9000/800 machines. This may foreshadow a good trend.

On a pro-certification note: if more vendors had small, easy to accomplish programs like Sun does, it would be easy for a system administrator to collect these from vendors whose gear they use. I wouldn't hold my breath waiting for a cooperative program from all the major players.

Just my rambling $0.02.

Opinion by Tony Sanders
<sanders@BSDI.COM>

My first impression is that the difference between doctors, lawyers, CPAs and system administrators is that their jobs don't completely change every two years and they deal with a consistent base set of problems.

Not so for the poor sysadmin, bombarded daily (it seems) with totally new hardware architectures, new operating systems, and new kinds of computing problems. The human body doesn't totally change every two years. Laws aren't written in Latin one week, Greek the next, then Esperanto (though it may seem so on occasion).

My second reaction is that certification programs are the wrong solution to the problem. I believe the Better Business Bureau is a better model to follow. A certification program replaces information with a warm fuzzy feeling and a piece of paper.

Certification could erode the quality of service because some businesses would hire people based primarily on the certification (the most likely trouble spots here are the largest employers). This already happens with college degrees ("Oh I went to XYZ state also ..."). It's too easy a filter with not enough meaning. I have personally witnessed this bias (you're a total wiz, get a degree or you won't be promoted in the company).

The counter to this trend is to create an elite class of certified people, which devolves into the AMA. Anyone care to explain why the AMA wants vitamins and herbs removed from store shelves? The AMA seems to desire legislation stating that they are the ultimate authority on the workings of mind and body. Bah humbug.

When faced with change, establishments tend towards protectionism, especially when they are wrong.

So You Want to Hire a Cheap System Administrator?

Opinion by Rob Kolstad
<kolstad@bsdi.com>

I wanted to share the stories of two different administrators that I've hired over the last decade to illustrate an important point.

I worked at a minisupercomputer vendor and had evolved into a position whose responsibilities included administration of over a dozen different timesharing systems. We were performing backups onto 9-track tapes at the time — and just changing the tapes on the systems required an inordinate amount of time. It was clear that extra help was needed.

My supervisor suggested that we hire someone who would be challenged by the task and could grow into a contributor in other areas as he/she gained experience. He suggested that I consider hiring a graduate of Control Data Institute or DeVry Tech (institutions that award the equivalent of an Associate Degree). After much searching, we ended up hiring Abel.

Abel was just over 30 years old and had recently changed careers in hopes of finding a position in the high tech industry. By no stretch of the imagination was he a programmer or even an engineer. Nevertheless, as an individual who had a dozen years of maturity beyond high school, it was anticipated that he would be up to the demands of performing backups in a reliable, timely, dependable manner.

Unfortunately for everyone concerned, Abel found himself surrounded by a number of hypercompetent engineers whose skills far outstripped his. Penultimately, he became insecure and fairly depressed. He worked on his backup-assistance program for months, ultimately creating 200 lines of code (with the help of tens of hours of other engineers' time). His impatience with himself grew as his self-confidence declined. His inability to grow frustrated him and everyone else. His attendance worsened slightly and he even was caught modifying dump logs when a tape error aborted a dump. He was too frustrated to re-run the broken dump.

Ultimately, Abel called me one Saturday evening and resigned in a drunken fit of frustration. After being convinced to defer the decision until Monday, he nevertheless stood by it and left the company. Abel was replaced by bright enthusiastic high school students who ran backups in the early evenings and generally did a far better job for far less pay. They fulfilled the earlier expectations of growth.

By way of contrast, I'd like to paint a picture that is an amalgam of several other administrators hired at the same site. Each of them used system administration as their stepping stone into the company. Each was extraordinarily competent and was able to soak up new knowledge and techniques like a sponge. I'll call the composite of these various administrators by the name of Baker.

Baker was someone who became a good student, sometimes later in his schooling. Full of enthusiasm, Baker had no idea which problems were 'unsolvable'. Baker spent his time automating tasks rather than performing them repetitively.

Baker acquired an Exabyte tape drive to solve the backup problem — thus relieving the burden of changing the dozens of 9-track tapes around the building (and saving money as well). Text processing software emerged in a workable form. Graphic previewers saved entire forests as technical writers were relieved of printing their intermediate drafts on paper (lots of money saved there, as well). User management systems, inventory systems, and even inventive CAD software emerged as problems were pointed out to Baker.

Eventually, the systems began running themselves. Baker, of course, was promoted into the software engineering group — only to be replaced by another bright administrator (a cycle which no longer may be seen as often in the challenging world of system administration).

Bright, skilled people have the knowledge and enthusiasm to wring the leverage from the powerful systems in use today. I am convinced that system administration should be performed by those with high levels of skill rather than those with the lowest — in spite of the pay differential and perception by those who only use the systems rather than manage them.

Review of BSD/386

"It works! It works!"

Reviewed by Lou Katz
<lou@metron.com>

Tuesday: FedEx package arrives. Open it. Very professional looking package inside. "BSD/386. March 1993. Version 1.0." Traditional shiny, slick paper. Inside of box is a CD-ROM with this weird gekko printed on it. Only one problem. I don't have a CD-ROM drive. Well, when all else fails, read manual. What do you know? It's very readable!

Wednesday: Order the latest and greatest CD-ROM drive. Then worry — will this system recognize it?

Saturday: Pick up new CD-ROM drive from El Cheapo Computer Company. Connect it to my SCSI controller. Boot machine. Well, it only sort of works. Try another SCSI address — 6! CD-ROM drive sighted by boot. Machine comes up and mounts drive! Decide to backup existing disks before proceeding further.

Sunday: Finish backup onto ancient, slow but trustworthy QIC-24 tape drive. Ancient, venerable machine, survivor of the infamous FaceSaver campaign is a 386-16 with 10 MB of memory. Machine currently has one MFM drive (about 100 Mb) and one SCSI drive (1.2 GB).

Now try to install this puppy. The goal is to have BSD/386 running without losing the ability to boot either DOS or Xenix from existing partitions. Anticipated the usual nightmares of incompatible disk partitioning schemes. After further reading of the manual, it was apparent that I would have to move the Xenix partition. Took all day.

Finally started the BSD/386 installation — it was a breeze! First convince DOS that I only had a 1 GB disk, and to use a partition at the beginning. Then BSD/386 used the rest, out to its real limit! Then load the base system.

Since I had the CD-ROM version and enough space, I just loaded almost everything and went to sleep.

Now to configure the system — lets see — need UUCP. Yup. But wait! My modem is not in the list of modems...ahhh...I HAVE SOURCE! Just like the olden days. Quick hack to put in my own modem's idiosyncrasies. Bidirectional TTY ports work fine. I need PCNFS. No problemo! Just RTFM and turn on the right daemons. Now I am a file server and my wife is happily working away on her DOS/Windows machine. PostScript printer which needed cat2dit to work with Xenix troff now up and running directly out of groff.

How about real adventure? Install SLIP/PPP mods to kernel. Kernel rebuilds right off of the CD-ROM by a neat hack. Bringing up PPP itself takes a little more work, mostly because the how-to's and why-for's aren't exactly clear in any book I could find. It now works, and I have my very own internet connection.

Import Eudora and POP. POP installs right away. Now mail can be read from my Mac (don't ask why!). Get a Mac-to-lpd utility. Mac printing spools through the BSD/386 lp spooler to printer. No longer have to push that dorky little switch on the back of the printer where I can hardly reach it (and can't see the interface number in the little window anyway) to go back and forth between local talk and parallel port.

Need to be able to convert AutoCAD plot files to EPSI form (PostScript with included TIFF preview image). No Problem! Get small utility from the net. Use ghostscript (provided) and pbmplus (provided). Hack-a-bit and there you are, thank you.

It is really a great relief having this system. It is even better than the good old days. First, anything I thought I might want seems to be there. Second, there is a VERY active mailing list which has an excellent signal-to-noise ratio and carries lots of good info. Third, the system is supported! Response to phone calls was good, though E-mail response to the reporting of bugs or problems was uneven. Unlike any of the other systems I have used (SunOS, Solaris, HP-UX, IRIX, Xenix, SCO UNIX, AIX, A/UX) there are no crucial missing pieces — no 'PostScript not included' nor compiler to be found in a separate licensed package.

I am hardly a speed or performance freak (with my antique equipment), but it seems that this system, under somewhat greater load due to the PCNFS functions is about the same speed as the Xenix system I ran on identical hardware. It seems to support enough of the mainstream peripherals so that I have had no problems with borrowed SCSI DAT drives as well as my old QIC cartridge drive. The system comes with X11, but I haven't exercised it yet, since I need a more reasonable VGA card first.

Besides all the utilities you would expect to find in a UNIX nowadays, as well as full, up to date networking support, there are also perl, elm, netfax, mh, TeX, nenscript, ispell, RCS, and access to DOS file systems on hard and floppy disks. There is enough interest on the net in this system that lots of software seems to come with BSD/386 as one of the possible compile options. AND THERE IS SOURCE (remember source?). If the man pages don't tell you what you want to know, you can always read it. And you can change it too.

This is not a perfect product, but in my environment it has been very stable, had all the features and functions I needed and does what I want. I would not hesitate to use it in a production setting, nor to install it on a client's machine. Some of the users have reported BSD/386 configurations running as network access servers with multiple dial-in lines, and as file servers. Unlike other commercial suppliers, the folks at BSDI have not gone crazy and have not priced a "PC" product like it ran on a mainframe. Further good news is that they expect to provide support for some of the binary formats of other systems in the (near) future. This would make it very attractive to configure, for instance, database and word processing applications in real commercial environments, because the clients could buy and use commercially and widely available packages.

Most of the problems I had were with the documentation. Many of the man pages were obviously the original BSD pages, and had not been edited to change path or file name references. Although one is supposed to be able to make changes to source and to compile a package from the CD-ROM, this only worked some of the time — the scripts to point to the revised source didn't always work. This is more of an annoyance than a fatal flaw, but it does waste some time. I eagerly await the 1.1 release, which may have some of the binary support and other neat features. If the BSDI folks put a reasonable effort into documentation and bug fixes this system could be around for a long time!

As Karl Malden might (but doesn't) say, "BSD/386, don't leave home without it!"

- BSD/386 V1.0 is available from Berkeley Soft ware Design, Inc.; 7759 Delmonico Drive, Colorado Springs, CO 80919; Phone: 1-800--4BSD, 1-719-593-9445; Fax: 1-719-598-4238; Prices for CD-ROM Source + Binaries $1045, Binaries only $545, price for Tape slightly higher. Version 1.1 is due to be released soon.

What's New

Editor's Note: As part of my ongoing quest to assemble quotes and short reviews of periodical literature, I approached John Gehl <gehl@ivory.educom.edu> *at EDUCOM about re-printing his twice-weekly blurb that summarizes information technology items. He generously granted permission. Edupage is a twice-weekly summary of news items on information technology provided as a service by EDUCOM — a consortium of leading colleges and universities seeking to transform education through the use of information technology. Let me know if you find this stuff useful or a waste of paper; write to* <kolstad@bsdi.com>.

You can subscribe to the full, long-version printed offering of EDUCOM REVIEW. For individual first-time subscribers, a one year trial subscription to EDUCOM REVIEW is $18 (a 70% savings from the regular subscription price). Send check for $18 to EDUCOM REVIEW, 1112 16th NW, Washington, DC 20036. Offer good only in US and Canada.

From radiation to rays. Samsung will begin marketing a "Bio-TV" next year that turns harmful electromagnetic radiation into ultraviolet and infrared rays, capable of making plants bloom and grow. (*Telecommunications Policy Review* 12/5/93 p.10)

Multimedia counterfeiting. An indictment in California marks what is believed to be the first counterfeit case involving CD-ROMs. The 29-year-old man allegedly imported illegal copies of CD-ROM products made by Software Toolworks Inc. Crimes like this cost U.S. computer companies more than $1 billion a year in lost sales. (*Wall Street Journal* 12/13/93 B4B)

Cyberspace market. The market for on-line services is estimated at $800 million in revenue a year, and is growing at 25% annually. (*Tampa Tribune* 12/13/93 B&F11) *Editor's Note:* I have attended presentations where this number is quoted as a "trillion dollar world-wide opportunity."

Telecommuting trends. One million more people are telecommuting this year than last, marking a 15% increase in company employees who work at

home part or full-time during normal business hours. According to a recent survey of more than 100 companies nationwide by *Home Office Computing*, 30% had some type of telecommuting program in place. (*Miami Herald* 12/13/93 p.24) A survey by *Work/Family Directions* found that 20% to 40% of employees surveyed would like to telecommute. (*Wall Street Journal* 12/14/93 B1)

Electronic yellow pages. Nynex announced it will put its Yellow Pages on the Prodigy network. Users will be able to scan listings and display ads, receive color photo images and pricing information on products, and check out restaurant menus and decor. (*Investor's Business Daily* 12/10/93 p.3)

Rules of the road. Twenty-eight companies have banded together to make recommendations to the Clinton administration on how best to achieve a seamless electronic superhighway. The Cross Industry Working Team includes AT&T, Apple, CitiCorp, BellSouth, IBM and Hewlett-Packard. (*Wall Street Journal* 12/13/93 B3)

Speedy LANs. Intel and SynOptics Communications will work together to develop hardware that moves information at 10 times the speed of Ethernet technology. (*Wall Street Journal* 12/13/93 B3)

MICROSOFT'S CHICAGO. Microsoft is rolling out the carpet for its new Chicago software, the next step up from Windows. Chicago will have a 32-bit data structure, be easier to operate, will have improved communications capability, and will include Plug-and-Play compatibility. "I will be surprised if at least 50% of the installed base of Windows users don't want to upgrade," said CEO Bill Gates. (*Investor's Business Daily* 12/13/93 p.4)

TI chips. Texas Instruments announced lab tests of a new chip that operates three times faster than the speed of conventional microprocessors at room temperature. The chips are based on principles of quantum physics, and use wavelength filters rather than traditional circuitry for directing the path of electrons. (*Wall Street Journal* 12/9/93 B4)

Next-generation Pentium. Intel has begun manufacturing the second-generation Pentium chip, and expects to ship between 2 and 7 million of them next year. The Pentium II operates at 100 megahertz and generates only 6 to 8 watts of power. (*Investor's Business Daily* 12/13/93 p.4)

Software piracy. The Canadian Alliance Against Software Theft lodged charges with the RCMP against three computer retailers under the Copyright Act. The retailers were convicted of illegally loading computers with copied software before selling them. Charges are still pending against a fourth retailer, a British Columbia college and a Toronto-based BBS. (*Toronto Globe & Mail* p. B5; *Ottawa Citizen* E2; *Ottawa Sun* 12/10/93 p. 52).

;login: 50 and 100 Years Ago

by Barry Shein
<bzs@world.std.com>

January 1943

The hope is, someday, of being able to deliver data between continents via peaceful uses of rocketry technology now so fearfully witnessed as effective in delivery of more odious goods in Europe.

Beating Swords into
Ploughshares
Babs Zwicky

With the Winter Conference almost upon us again we wish to remind our membership that we are collecting restaurant suggestions in the Hoboken area. Of particular interest are restaurants which serve meals prepared from non-rationed foods.

Letter to the Editor
Lefty Tower

January 1893

Rumors abound concerning a method for showing "moving pictures" being developed in Thomas Edison's New Jersey laboratory. Although quite secretive on the details, Mr. Edison has suggested a presentation at an upcoming USENIX conference.

The Industry Companion
Hortense Mark

President Cleveland has graciously agreed to present the keynote address at our upcoming Washington, DC, winter conference. *The Farmer's Almanac* predicts some snow is possible for that week, so bundle up and be sure to join us!

Executive Director's Report
Effie Young

Imake *Rhymes with* Mistake

Controversy by Jeffrey S. Haemer
<jsh@canary.com>

I said, in a review I wrote for this newsletter that *Software Portability with imake* [Paul DuBois, 1993. *Software Portability with imake.* O'Reilly & Associates, Inc.] is a fine book: well-written, well-edited, and useful. Here, I'll argue that it's a fine book about a dreadful idea: *imake.* Actually, I've seen enough *imake*-like *make* front-ends to suspect that such things are as common as peas. I don't like any of them. Why?

Let's begin with an example from the book. Here's a Makefile for the "hello, world" koan.

```
hello: hello.o
       cc -o hello hello.o
```

With this, we can say

```
$ make
cc -c hello.c
cc -o hello hello.c
```

By page 30 or so, we discover that *imake* lets us transform this into the more portable, more flexible Imakefile,

```
SRCS = hello.c
NormalProgramTarget(hello,hello.o, NullParameter,NullParameter,Nullparameter)
DependTarget()
```

which lets us build our program with the sequence

```
$ make Makefile
$ make depend
$ make
```

Yes, it looks like PL/I or X, whichever you hate most. Yes, the whole `NormalProgramTarget()` call has to be on one line, which has required ORA to invent a typesetting convention to show continuation lines in a way that doesn't suggest you could actually use continuation lines yourself. Yes, it's noticeably longer and more complex than what we started with. And no, you can *not* abbreviate the first two lines as `$ make Makefile depend`, the way you'd expect.

By the way, *imake* is invoked by the makes shown above. You could invoke *imake* by hand, but that'd be a lot uglier.

Oh, and what I said about making targets from the Imakefile isn't quite true. Typos: for example if I mis-typed the third `NullParameter` above, turn the process into

```
$ make Makefile # imake failure,
       #which steps on the
       #existing Makefile
$ vi Imakefile # fix the problem
$ xmkmf # regenerate, default Makefile
$ make depend
$ make
```

You'd need at least a second iteration of this after you discover that the blank following "`hello.c`," in the argument list is also impermissible.

"But that's not fair!" you object. "Imake isn't designed to make simple tasks simple."

Exactly. So what problem is it designed to solve? Some people use it for making X. Before analyzing what other nails this hammer might be suited for, let's step back a second to look at the problems standard *make* doesn't solve.

A vice-president of the late, lamented INTERACTIVE Systems Corporation, once told me that roughly half of all changes in a largish SCCS tree he'd inspected were in the Makefiles. (Yes, Virginia, some vice-presidents can inspect SCCS trees. And you thought it was nice that Al Gore could spell.) Why? Makefiles change proportionately often for four reasons, the first of which is that *make* wasn't designed to handle many of the sorts of changes that ports require. It has no `#include` facilities, no `#ifdefs`, and no real way, within the language, of defining procedures.

One tempting work-around is to build a thick front end to *make. Imake* uses *cpp* and a slew of configuration files, but I've seen other solutions — with *m4*; with elaborate shell scripts; with an `include` extension to *make* and obligatorily included `*.inc` prologues and epilogues — that differ in detail but not in philosophy or final effect.

What goes wrong with such solutions?

- Suitability. The solutions, designed with one problem in mind, don't quite fit other problems.

Imake's `LintTarget()` rule, for example, lints all sources at once, whether they go into the same executable or not; an *imake*-generated Makefile always lets you do a `make clean` — which removes `*.BAK, .emacs_*`; and so on.

- Portability. Not code, people. Even most undergraduates can read and write Makefiles. Learning a thick front-end becomes learning a new language, which can eat up all the time you thought you were saving and make turnover very expensive.

- Debugging. Trying to figure out what's wrong with a Makefile is hard enough. Trying to figure out what's wrong with an unfamiliar language that's processed into an unreadable Makefile (an *imake*-generated Makefile from even a trivial Imakefile is hundreds of lines long) is downright tedious. Okay, okay. This probably isn't a problem for you because your code always works the first time.

- Speed. The author gives no performance figures, but I've personally sped up builds by a factor of 80 (really) in one such environment by abandoning the front-end in favor of a clean make scheme with simple Makefiles. I've watched impatient developers in environments with heavyweight make-front-end schemes cheat, time and again, with hand-crafted Makefiles. Worse, I've seen developers, unwilling to put up with the multi-step process, tinker directly with the machine-generated Makefiles and, later, archive the results under SCCS.

- Over-generalization. *make depend* insures that each file depends on its included headers. This means that every build checks the time stamps of every file against all of the included system files, like `<stdio.h>`. This is almost always the wrong thing to do, which is why you and I seldom include those files in our dependency lists when we write Makefiles.

- Maintenance. *Imake* includes a utility called *mkdirhier* that creates directory hierarchies because some *mkdir*s don't support *mkdir -p*. Today, the right solution is to supply a *mkdir* that does. Sources for such *mkdir*s are publicly available. Indeed, any *mkdir* that doesn't support *mkdir -p* is not POSIX.2-conforming, so supplying a conforming *mkdir* would be a favor to everyone. I'd bet that no one will make this change to *imake* soon. Changes to any bulky, complex, widely distributed piece of software are undertaken with great care because of the risks of unexpected side effects.

Still, the bigger the problem, the more you can afford to invest in special-purpose tools to save you time in the long run. X is clearly a big problem. On BSDI, for example, the X11R5 sources are as large as all the rest of the operating system — kernel + utilities — put together. And, of course, X11R5 is intended to be built on an unmatched array of UNIX and non-UNIX platforms. If you have a problem this big, maybe *imake* is right for you.

Or maybe not. "What would *you* recommend?" you're asking.

First, for really big problems, why not go whole-hog? If you're going to invest in a tool, look at a *make* replacement, like Andrew Hume's *mk* or Glen Fowler's *nmake*. These offer more power and flexibility than *make*, plus hefty performance enhancements, but preserve *make*'s general model and syntax. You can read and write Makefiles for these next-generation tools without hours of training and weeks of practice.

Medium-sized projects, where you can't afford as large an investment in tools and training, call for a simpler solution. Many publicly-available *make*s, such as GNU's *make* or the *make* from the Berkeley Net-2 release, supply enough extensions to handle the problem. One of these, perhaps combined with a thin layer or two of shell scripts, such as those described in Lapin's aging, yet still useful, *Portable C and UNIX System Programming* [J. E. Lapin, 1987, Prentice-Hall], should do the trick. (I haven't yet played with it myself, but for tougher domains of portability, FSF's autoconfig scheme looks promising.)

For small projects, use *make*, and learn how to write good Makefiles. At one point, when describing a difficulty that arises in handling multi-directory projects with *imake*, DuBois remarks, "Fortunately, many projects consist of only one directory anyway." In such cases, which would you bet is faster: learning, using, and configuring *imake*, and then writing and maintaining the Imakefile, or just porting your Makefile along with the code? To improve your odds, keep your Makefiles simple, clean, and easy-to-read. I often hear the advice, "Start with simple Makefiles, then work up." That's usually half right.

Imake *Response*

by Paul DuBois
<dubois@primate.wisc.edu>

General remarks

Haemer suggests some alternatives to *imake*, e.g., a new *make* program with which the reader may not be familiar. That's useful, since some readers may try these other tools and find them suitable for their work. However, the discussion would be more compelling if he named some actual projects that are configured with these tools, especially for those intended for large projects. And in discussing tools for medium-size projects, Haemer says "one of these, perhaps combined with a thin layer or two of shell scripts ... should do the trick." I'm left thinking, "will they or not?"

imake is known to be capable of configuring X, Motif, Khoros, Andrew, and more. All these are empirical demonstrations of its utility. Such are missing for the alternative tools.

One theme that emerges consistently throughout the article is that we should use simple, clean Makefiles. I'm all for that. Unfortunately, my experience suggests that simple Makefiles are insufficient for all but the most trivial programs, and require per-system editing otherwise. Haemer gives me little reason to believe otherwise, beyond general statements that certain alternatives offer "power" or "flexibility." Maybe, but do they really help me write portable software? Do they solve the problems of suitability, portability, and debugging that attend the use of *imake* and related programs? If so, how? I'm left wondering.

Specific comments

In the discussion of the "hello, world" Makefile: "By page 30 or so, we discover that *imake* lets us transform this into the more portable, more flexible *Imakefile*..." I suppose the intent of the "by page 30 or so" wording is to convey that it takes a lot of work to get to the point where we can write an *Imakefile* for even the most trivial program. If so, it's misleading. One purpose of the "tour of *imake*" chapter in which the example appears is to provide a lot of explanation to help the reader understand what's going on. And even with the explanatory text it doesn't really take 30 pages to get to the *Imakefile*.

The *Makefile* appears on page 19, the *Imakefile* on page 26. Haemer's narrative overstates the argument, by making it seem worse to write an *Imakefile* than it is.

Haemer comments that the *Imakefile* is longer and more complex than the hand-written *Makefile*. And that's true — for the example he shows. But the *Imakefile* shown is a special case in a contrived tutorial situation. It was written as it was so it'd be easier to extend to the more complex situations covered later in the chapter. Were one to write the optimal *Imakefile*, it would look like this instead:

```
SimpleProgramTarget(hello)
```

Which is shorter and simpler than the *Makefile*. In any case, we're talking about a toy program. Once a program becomes non-trivial, portability becomes an issue because you quickly run into *Makefile*-editing uglinesses. As complexity increases *Makefile* portability decreases. This is not true of Imakefiles.

Haemer also discusses problems that occur when you make errors in your *Imakefile*. The concerns Haemer raises are valid, although again I believe he overstates his argument. He lists a sequence of commands you go through to fix a botched *Imakefile* and build your software. But is the situation better if you write a clean, simple*Makefile* directly? You can still make mistakes, and if you do, you still need several commands to fix the *Makefile* and build the software. And then you get to edit it each time you move the software to a different machine. At least with *imake*, you don't have to change your *Imakefile* once you get it right.

The discussion about extra blanks in rule invocations could be stronger. It's important to avoid extra blanks in rule arguments because they *may* cause problems, not because they inevitably *will* cause problems. The extra blank inserted by Haemer is benign and the *Makefile* will still be built correctly, so the example is unconvincing. A stronger case would be made by an example where a blank really does break the *Makefile*, e.g.:

```
SimpleProgramTarget(hello )
```

About 'suitability': "The solutions, designed with one problem in mind, don't quite fit other prob-

lems." Well, yes. But that's true of any solution to any problem.

It's unclear what the point of the comment about the *make clean* command is —that it's too broad and removes too many different kinds of files? This could use some clarification.

Portability: "Not code, people." Huh?

Debugging: "Trying to figure out what's wrong with a Makefile is hard enough." But the previous bullet item says that even most undergraduates can read and write them. There's an inconsistency here that should be eliminated to make the overall argument more cohesive. The point about trying to figure out an unreadable *Makefile* has some merit, although I disagree that it's so bad as all that. Most *Imakefile* problems manifest themselves such that figuring out the problem from the broken *Makefile* is reasonably straightforward (as is mentioned in Chapter 6).

Speed: Haemer says simple Makefiles speed up builds a lot. I'm sure they do, but complex projects usually require more than simple Makefiles, which then are not portable. Anyway, there is such a thing as running processes in the background or in another window while you work on something else.

I was confused by the 80x speedup comment; maybe some other readers are, as well.

I'm sympathetic to the plight of the poor souls Haemer describes, but I'd say that in the long run, or for large projects, the right way to proceed is to get *imake* running correctly than to mess around with zillions of Makefiles. My observations suggest that the worst part of using *imake* is getting it built and installed. That's a real problem, but you only have to do it once. After *imake* is installed, building *imake*-configured projects typically proceeds without incident. Difficulties that do arise often tend to involve non-portabilities in program source code rather than the configuration files, and you can't improve badly-written source code by writing good Makefiles. Or, indeed, with *any* configuration system.

Over-generalization: Perhaps it's true that in some situations you don't want all the system header files listed among the dependencies. In my environment I certainly do want them. In any case, I don't see that it does any harm to have them listed. So is this really an issue? If it is, list dependencies as you like manually in your Imakefiles and put this line in *site.def*:

```
#define DependTarget() depend::
```

Maintenance: I'm not sure what is being "maintained" here, or what, exactly, is being claimed. The argument seems to be that one program (*mkdir*) can be supplied rather than another (*mkdirhier*), and I don't see what difference it makes.

If *mkdir -p* is indeed equivalent to *mkdirhier*, then by all means go ahead and modify your configuration files to use it. That isn't difficult, and *imake* doesn't restrict you in any way from using the one you prefer.

Conclusion

I don't want to defend *imake* as being the most elegant solution ever to the portability problem. It's not. It's a crude, ugly, hack and is widely acknowledged as such, e.g., by its author and by the people that wrote the X11 *imake* configuration files.

Nevertheless, *imake* survives because it works and it's useful. Some of Haemer's criticisms are valid, but he hasn't convinced me that the alternatives he cites are going to save me any work. Perhaps he would, were more concrete specifics given.

The Fifth International Olympiad in Informatics

by Donald T. Piele
<piele@cs.uwp.edu>

Editor's Note: Don Piele is the USA Team Leader for the IOI, which is the international high school programming contest. USENIX is a co-sponsor of the USA rounds of this contest.

Our adventure began Friday, October 15, 1993 at the Miami International airport where the USA Computing Olympiad team met for the first time since the summer training program at the University of Wisconsin-Parkside. Dr. Harold Reiter, the deputy team leader, flew back from London where he was spending the year teaching mathematics at Kingston College. Team member Hal Burch, 18, flew in from Missouri where he is a freshman at the University of Missouri at Rolla having graduated in June from the Oklahoma School of Science and Mathematics in Oklahoma City. Eric Pabst, 17, came from Salt Lake City, Utah, where he was a senior at East High School, and Mehul Patel, 16, arrived from Houston, Texas, where he was a senior at Langham Creek High School. Yonah Schmeidler, 17, a graduate of Ramaz School in New York and now a freshman at MIT, had flown earlier to Buenos Aires and would meet up with us on Sunday. Our next stop would be Santiago, Chile with a connecting flight over the Andes mountain range to Mendoza, Argentina, the site of the Fifth International Olympiad in Informatics (IOI).

We left the United States at the peak of the fall colors and arrived in Mendoza in the full bloom of spring. We were met at the Mendoza airport by a contingent of college students from Mendoza University whose job for the next ten days would be to guide the participants (273 students and team leaders from 45 countries) to various events within the city and one excursion to the Andes. Their enthusiasm and warmth was infectious.

Eric savored the opportunity to try out the Spanish that he had studied for five years, and he quickly established a special relationship with our hosts. His facility in the native language proved to be a big asset for him as well as all the members of the U.S. team. Several times during our stay he would be called upon to give radio and T.V. interviews, talk with the press, help us translate stories that appeared in *Los Andes*, the local newspaper, and find the beef and chicken dishes in a restaurant menu.

IOI participants were housed in two hotels, and our team stayed at the Hotel Aconcaqua, named after the highest mountain in the Western Hemisphere (near Mendoza). Fifty Compaq computers were set up on the hotel's second floor for students to use. Similar Compaq computers were housed in the Convention Center, approximately six blocks south of the hotel, where we had our meals and where the competition was held.

The city of Mendoza had purchased 400 Compaq computers for the event which would be used by the city when the competition was over. There were enough computers around to completely outfit the team leaders' room with a networked system complete with e-mail and printing capabilities. This was a first for IOI and a very appreciated feature of this years' Olympiad. I used it to keep in touch with family and supporters back home.

On Sunday evening, we all gathered at the Convention Center for the opening ceremonies. Argentinean officials, including the director of technology in education, the mayor of Mendoza, and the 1993 IOI organizer, Dr. Alicia Baquelos, gave their addresses in Spanish, which were translated paragraph by paragraph into English, the official language at IOI. A festive mixer erupted soon after with a Latin beat drowning out any attempt at conversation.

Tuesday was the first day of competition. The team leaders and deputy team leaders were given a wake-up call at 3:30 a.m. so we would be ready for the early morning jury meeting at 5:00 a.m. in the Convention Center. The main order of business was the selection of three problems for the first day's competition. These were selected from a set of nine problems submitted by the scientific committee from Argentina. Besides choosing the problems, all of the non-English speaking countries needed to translate the problem statements into their language and have copies made for each of their participants. There were approximately 35 different native languages represented, and everything needed to be ready at the appointed starting time.

The competition began at 11:00 a.m. and 155 students each went to their personal Compaq 386 machine in one of four different rooms and "started their engines." They had five hours to solve three problems, using one of the officially installed languages: Turbo Pascal V. 6.0, Turbo C++ V 2.0, Quick Basic V 4.5, and LCN Logo V 3.0.

The sponsors and coaches had been working for 6 hours straight in a smoke-filled room, and it was time for a much deserved rest. But before we could relax, we had to remain on call to translate any written questions the students might ask during the first hour. Then we were free to go and get some rest before the judging began.

At 4:00 pm the competition ended and the students filed out of their rooms with looks of confidence and relief. For the next several hours they would be called back, one at time with their team leader, to have their programs checked by a local coordinator who had been trained to run the programs against a series of input data and evaluate the output file for the correct results. If all runs were perfect, the program was awarded 100 points. Hal and Mehul's programs were flawless and Eric and Yonah's were close behind with 71 and 62. No scores are officially posted for the first day but we quickly learned through word of mouth that a total of 16 students had perfect first round scores.

The next day, Wednesday, was reserved for touring a local chocolate factory, followed by a barbecue at the country home of one of the organizers of the IOI. One of the special treats in Argentina is to cook large hunks of fine beef very slowly over an open pit. The meat is then sliced off and placed on buns and topped with a special mustard sauce. This makes an excellent sandwich, and the lifetime of each platter full of meat could be measured in nanoseconds. After a delightful afternoon, we returned to our hotel to get ready for the final round.

Thursday morning began at 3:30 a.m. and was a repeat of Tuesday, except that this time exactly one more difficult problem was selected from a set of three. One problem was eliminated because of its ambiguous wording and the difficulty of making is completely clear in 35 different languages. Almost any problem can have different interpretations depending on how it is translated. The problem we chose was clear, but one thing we forgot to discuss was how the solutions to this problem would be graded. This, unfortunately, led to a major misunderstanding.

This problem quickly became apparent when we walked into the computer room with the coordinator and saw for the first time the rules used to judge the eight sample runs. The first six data sets had a limit of two minutes and the last two a limit of five minutes. Everyone on our team had solutions that ran instantly for the first seven data sets but all ran over the five minute limit for the last and most difficult data set. Since this run was worth 25 points, their hopes for a gold medal vanished as did the hopes for 12 other participants who had perfect scores the first day and also did not optimize for speed. They had fallen into the exponential time trap which for many could have been avoided had they known that, for the first time at IOI, speed would be the deciding factor.

Last year in Germany I was surprised to learn that the speed and efficiency of an algorithm were not considered a factor in grading. In fact, several programs were allowed to run for hours, even overnight, and others finished in seconds; yet no distinction was made between them. I thought this was rather odd, but everyone seemed to accept this as an unwritten rule of IOI. Students were aware of this fact, and we had told our team members to play it safe and go with any working algorithm and not to worry about speed unless it was explicitly stated in the problem.

It never occurred to the jury to ask how the problem would be graded, and when the situation surfaced after the competition was over, it was too late to correct. Many students were well aware that their programs could take years to complete if a large number of data points was used as test data, but since time had never been a factor before, they thought it would not be a factor here.

But this was not to be. The jury reacted to this situation by drafting additional competition rules to be considered for IOI'94, including: "When a time limit will be applied during evaluation, it should be explicitly stated in the problem description." Had this been done at IOI'93, it would have helped a great deal. Of course, speed of execution as a factor in the grading of solutions is not a bad idea. Since this was the first Olympiad to breach the time barrier, it will now be on the minds of all team leaders as they prepare for IOI'94. The omission, of course, affected everyone equally.

A long bus excursion into the mountains of the Andes was reserved for Friday. Our final destination was Uspallata, a ski resort high in the mountains. Here we were treated once again to the famous open pit beef barbecue done on a grand scale. Inside the dining hall, the participants were happy grazing on all the beef they could eat and toasting a local guide whose birthday had been

discovered. We soon had to leave to get back to an important jury meeting to decide the cut off scores for the gold, silver and bronze medals.

Back at the convention center the jury met to decide who would get the medals. According to the rules, only half of the students, can receive a medal. This rule helps maintain the value of each award. Also the gold, silver, and bronze awards must be given out in the ratio of 1-2-3 (or as close as possible). Out of a possible 200 points it worked out as follows:

Points	Award	Number
180-200	Gold	131
60-179	Silver	27
125-159	Bronze	39

Hal Burch and Mehul Patel received silver medals and Eric Pabst and Yonah Schmeidler received bronze. Our team ranked 7th out of 45 in the total number of points and for the first time two girls won silver medals, one from the Czech Republic and one from the Slovak Republic.

The seven teams that won four medals were:

	Points	Gold	Silver	Bronze
Slovak Republic	714	2	1	1
Romania	691	2	1	1
Russia	683	1	2	1
Iran	660	1	2	1
China	644	1	1	2
Korea	640	1	1	2
USA	633		2	2

Gold medals were also won by students from: Sweden, Czech Republic, Bulgaria, Belarus and a United Nations team from Yugoslavia.

After the jury adjourned, the U.S. delegation was invited to attend a meeting of the International Committee to see when we would be interested in hosting an Olympiad. Countries that had submitted proposals up to 1997 were: Sweden-1994, Netherlands-1995, Hungary-1996, South Africa-1997.Several countries were invited to this meeting to announce tentative plans to submit proposals for years to come. They were: Portugal- 1998, Turkey-1999, China-2000, Thailand-2001, Korea-2002. We were also interested in the year 2000 but since China had been a member of IOI longer, they were given precedence over any proposal from a newer member.

The awards ceremony was held on Sunday and began at 9:30 a.m. at the Independence Theater. All medal winners were seated on the stage, with the delegates, other participants and spectators seated in the audience. After the opening ceremonies each team leader was invited to the stage to present the medals to their team members, starting with the bronze and ending with the silver. For the gold medal winners, the students received their award and prizes from local dignitaries from Argentina. The top four students, who were tied at 200 points each, received computers and they were awarded a new IFIP trophy that will go each year to the top student or students in IOI.

Pictures were taken as the trophy was hoisted into the air by four excited and deserving young men from The Czech Republic, Romania, Iran, and Sweden. The torch was passed to Sweden who invited us all to the 1994 IOI in Stockholm, and the curtain rang down on another successful International Olympiad in Informatics.

Thank you, Argentina.

An Update on UNIX-related Standards Activities

by Nick Stoughton
USENIX Standards Report Editor
<nick@usenix.org>

A standards committee was formed to develop a new standard for Open Systems. The project was approved and the committee got down to work.

For forty days and forty nights the standards committee ate nothing, but wrote their standard. They became exceedingly hungry. Then the devil appeared to them and tempted them to get food by going to ballot early. "It will prove you truly are a great standards committee," he said.

But the standards committee told him, "No! For it is written that bread will not fill a standards writer's soul: obedience to every word of the procedure is all we need."

Then the devil took the standards body to a great International Organisation, and said "If you rewrite your standards in a computer Language Independent form it will prove that you are truly a great standards organisation. Angels will appear to prevent you from smashing on the rocks below." The standards committee retorted "It is also written that existing practice shall be followed, and there is no existing Language Independent practice to follow."

So the devil took the standards committee to the peak of a very high mountain, and showed them the governments of the world in all their glory. "Every one of these governments will require all their people to adopt your standard, if you will worship me and be prepared to invent a set of new communications protocols."

But the standards committee said "Get thee behind us Satan. The procedures say follow only existing widespread practice. Obey only the IEEE."

Whilst some liberalization of the facts has been used to make them fit the story above, all these things have happened over the past few years within several standards bodies.

- The Language Independence Issue raged within the POSIX world for two years or so, until last summer, when, finally, the IEEE agreed to drop the requirement of ISO that all new and existing standards had to be written in a language independent form. This would have meant, for example, that the existing ISO 9945-1 (POSIX.1), which was written using the C programming language, should be rewritten. As suggested by the parable above, this was viewed by most people within POSIX as akin to taking yourself to the top of a tall building a jumping off. When Jesus was tempted in the wilderness, I am sure He had a far higher degree of certainty of survival if he had thrown himself from the pinnacle of the temple. For POSIX, the choice was between following the mandates of ISO, taking forever to produce a standard that no-one could understand or use, and ignoring ISO, thereby risking international acceptance and status for the resulting, language dependent standards.

The third temptation in the parable is probably the most interesting. Why shouldn't Open Systems Standards be invented? The old saying "You can take a horse to water, but you can't make it drink" springs to mind. Good standards are ones that people will want to use. Bad standards, even if they are mandated (e.g. the set of OSI protocols selected for GOSIP), will never gain widespread acceptance. When making a standard, most bodies look around to see what everyone is doing in the area in the lack of a real standard. Big companies, like Microsoft, say "We are so big and powerful, we'll do our own thing and everyone will follow, because we have a good marketing department!" More formal standards bodies document existing practice, in formal language. Sometimes a little massaging is needed to fit together the pieces in a smooth fashion; occasionally there is a glaring hole discovered by the process that a small invention could be allowed to cover. But there lies a slippery slope. Once you allow one little bit of invention, you allow another, and another, till there's little left of the original base document.

Standards bodies are made up of technical people, knowledgeable in the specific area they are standardizing. So why can't they invent new things in their area? Why can't Microsoft rule the world with Windoesn't (Windows NT)? What was wrong with the OSI protocols? Well, probably the single biggest thing is that Internet Protocol Suite, including TCP/IP and all the related protocols, is a very low cost, higher performing, and embodied in an enormous existing network. True, OSI and TCP were being developed concur-

rently, and at the time OSI was being made, not all the above were true! Nevertheless, there was enough existing practice to show that TCP/IP was going to succeed. What OSI produced looks good on paper, at a high level. But what people were doing at the time was ignored. Apart from Government applications, OSI is rarely in use, whilst the Internet, well, need I say more...?

Report on POSIX.4: Real-time Extensions

Lee Schermerhorn <lts@westford.ccur.com> reports on the October 18-22, 1993 meeting in Bethesda, MD:

The POSIX.4 Working Group Chair was unable to attend the meeting because of "real work" commitments. The Vice Chair was also in absentia because of imminent fatherhood, of which there seems to a lot going around lately. The forced absence of the Chairs left the running of the meeting to the "third string"— the Secretary of the Working Group who happens to be your POSIX.4 snitch reporter for this meeting.

So here's the plan: first we'll review the status and schedule of the documents that have already been reported out of the working group for balloting; then we'll cover the activities of the Working Group during the week.

Balloting Status and Schedule

- POSIX.4 **aka** POSIX.1b: It's official. The IEEE Standards Board approved Draft 14 of the POSIX.4 Realtime (one word according to POSIX.4) Extensions Standard at the mid-September meeting. At nearly the same time, the IEEE was also renumbering the standards to confuse the innocent. Because POSIX.4 is cast as modifications and additions to POSIX.1, the IEEE has renamed POSIX.4 to POSIX. 1b. Sort of makes sense, except that POSIX.1b will be published well over a year before POSIX.1a! So it's best not to think of the letter suffix as a revision.

- It appears that POSIX.4/1b will be published as a merged document to replace the current POSIX.1-1990, in the March timeframe. In the meantime, the full Draft 14, as opposed to the small set of changes that were actually balloted in the last recirculation, is available from the IEEE at a "modest fee."

- POSIX.4a **aka Pthreads aka** POSIX.4c: Draft 8 of Pthreads is being recirculated for a 10 day ballot period from November 1-12, 1993. "Recirculation" means that only the changes from Draft 7 are open for comment and/or objection. JohnZ, the Pthreads (and POSIX.4) technical editor, expressed the opinion that one additional recirculation will be required to clean up loose ends. This would make it unlikely that Pthreads can be ready for the March 1994 Standard Board meeting. The June 1994 meeting is a more likely target.

- Note that POSIX.8, Transparent File Access (TFA), is also expected to be approved at close to the same time. The System Interfaces Coordinating Committee (SICC) has noted this and has determined that Pthreads will be merged with the then merged POSIX.1/1b standard before TFA. It remains to be seen when, and in how many volumes, the results will be distributed.

- POSIX.4b **aka** POSIX.1d **— more Realtime extensions**: Draft 8 of this document was reported out of the working group for ballot again in July. The first ballot is open for 30 days starting on 1 Nov 1993. Those of you who follow comp.std.unix may recall that a call went out for all the UNIX true believers to join the balloting group to make sure that those wild and crazy POSIX.4 real timers don't do something unclean (in the UNIX sense) to POSIX.

- POSIX.4b/1d contains several additional real time extensions, including:

- The *fadvise()* file advisory chapter that replaces the "real time files" chapter that was removed from the last draft of POSIX.4.

- A "Sporadic Server" chapter for budgeting CPU time to aperiodic events so that they can be handled via Rate Monotonic Scheduling analysis, with guaranteed deadlines.

- Definition of Process Virtual Time Clocks under the POSIX.4 Clocks and Timers interface. These are analogous to the virtual "itimers" of BSD and SVr4, and are included primarily in support of the Sporadic server.

- "Device Control"— really *ioctl()*, but with some "enhancements" to address some standards/portability related issues that kept *ioctl()* out of POSIX.1. Wouldn't it be nice, if before the balloting is over, this ends up as good old *ioctl()*?

- "Interrupt Control"— connection of user programs to interrupt sources. Two modes of operation: one where an application requests notification via signal when a particular interrupt occurs — without having to write a driver — and one mode where a user specified function is run at interrupt level. I suspect this one will have a lot of difficulty in balloting.

- POSIX.13 — **Realtime Application Environment Profiles**: It is over a year since the first

round of balloting on the POSIX.13 profiles closed. Ballot resolution has been slow because of three gating issues:

- The POSIX.13 Profiles reference the POSIX.4 and POSIX.4a Draft Standards and would, in any event, have to wait for both of these Standards to be approved.
- The POSIX.13 Draft contained four (4) profiles in a single document. An earlier draft of the ISO document that defines profiles (TR 10,000) apparently forbade multiple profiles in a document.
- Three of the four POSIX.13 Profiles restrict an application to a subset of the interfaces in POSIX.1. PASC Profile Steering Committee (PSC) rules for profiles — non-existent when POSIX.13 first went to ballot — forbids a profile to specify a subset of a base standard.

The first roadblock is in the process of resolving itself. POSIX.4 is a done deed, and Pthreads should be approved by mid-'94. A later draft of TR10,000 now allows multiple profiles in a "Standard Profile", if a number of conditions regarding cohesiveness, etc. are satisfied.

The final issue is one which has consumed vast amounts of PASC meeting time, in Working Groups, PSC meetings, SEC (Sponsor Executive Committee) meetings, and in hallway/bar room conversations. An intensive effort during the week of meetings by an Ad Hoc of the SEC has resulted in a compromise, of which more later.

- **LIS — RIP Or "What ever happened to** POSIX 4c?" POSIX.4c was to be the Language Independent Specification (LIS) of POSIX.4. But when, in July, the SEC rescinded the requirement for Working Groups to produce LIS for all PASC Standards, the POSIX.4 Working Group immediately voted to stop work on their LIS. That decision was confirmed again at this meeting.

  Thanks to the efforts of Michael Gonzalez, the Working Group has a nearly complete first draft of the LIS. Michael said that he wanted to complete the remaining couple of sections, and would like to see the results be made available to anyone interested. The WG has been assured that it will be no problem to arrange to have the completed, unreviewed draft available for ftp from both the IEEE's emerging SPA (Standard's Process Automation) system, or from Michael Gonzalez's University system at University of Cantabria, Santander, Spain.

Working Group Actions and Plans

With all of its documents, except for POSIX.13, done or out for ballot, one might well wonder what the POSIX.4 working group is doing meeting in exotic places like Bethesda, MD. Two things: planning for additional drafts to standardize additional interfaces, and POSIX.13 ballot resolution.

First, POSIX.13 ballot resolution: The Profiles ballot resolution effort had degenerated to getting the issue of specifying subsets of POSIX.1 resolved. Because this issue is one of inter-Working-Group coordination, it required a lot of interaction with members of the ad hoc committee established to report back to the SEC. Several members of the Working Group, who are also POSIX.13 technical reviewers, — Andy Wheeler, Joe Gwinn, and others — spent a couple of hours every day, Monday through Thursday, in the ad hoc; reporting back to the Working Group daily on progress or the lack thereof.

The ad hoc made a fairly thorough review of the issues, noting that the primary objection to the subsetting was more religious and political than technical— that is, the "dilution" of the POSIX name if it were associated with anything less that full POSIX.1-1990 as we know and love it. In truth, though, a number of technical issues did surface concerning testing of subsets, the effort of respecifying the semantics of POSIX.1 with formal subsets, the integration of Standards that later modify the full POSIX.1, such as Pthreads, POSIX.8, etc., with a subsetted POSIX.1.

Ultimately, the ad hoc placed a resolution before the SEC to suspend the PSC rules for the "special case" of real time subsets for POSIX.13, and allow POSIX.13 to specify the subsets in the profiles. After an hour and a half of debate in the SEC, the motion passed, with an amendment requiring that the POSIX.13 balloting group be reopened for a minimum period of 30 days. The primary objections to the motion were not in objection to allowing POSIX.13 to subset POSIX.1; so much as to having the subsetting done in POSIX.13. The view was that if subsetting were to be done, do it once and for all in POSIX.1. This would probably hold up not only the POSIX.13 profiles for a couple of more years; but any extension standards that happened to coincide with the subsetting revision. The resulting resolution will provide the embedded real time systems community — users and vendors alike — with a standard profile that describes the runtime environment that the target applications can depend on, and that conforming

implementations must, as a minimum, support. The Chair of the SEC pointed out that later, when extensions to POSIX.1 settle down, and the real time (subset) profiles have had some use, might be an appropriate time to formalize the subsets in the POSIX.1 standard itself.

The SEC resolution now clears the way to complete the POSIX.13 first round ballot resolutions. But, a fair amount of work now falls on the Technical Reviewers to add the normative text that effects the subsetting to the next Draft. A not so small group of volunteers signed up to work on and review drafts of the subsetting text. The approach discussed in the Working Group is to prescribe what functions are available to Strictly Conforming Applications for each profile. Where some subset of behavior of a required function is not required, it will be explicitly unspecified. For example, *open()* of a non-existent file in a profile with no requirement for a file system will be unspecified; rather than, say, return a specific error. Initial drafts should be available for the January meeting.

The other new work item was additional interfaces for — call it POSIX.4d. The Working Group has had a running list of features and functions of real time systems that are potential candidates for future Real Time extensions of POSIX.1. But, the Chair has instructed the Working Group that we won't generate another PAR unless concrete proposals, including base documents, are presented, backed by a strong commitment to see them through to standardization. The Working Group reviewed several proposals, a couple of which had fallen out of earlier work such as POSIX.4b because of lack of consensus at the time that '.4b was otherwise ready for ballot. The new proposals include:

- **Typed Memory**: This is essentially an additional type of memory object, like /dev/mem, that represents different views of special physical memories, such as external memory modules visible on multiple busses. Extensions to *mmap()* support additional flags for dynamic/contiguous allocation by the object and functions to obtain an offset within the object from the address returned by *mmap()*, needed with dynamic allocation.

- **absolute nanosleep()**: This is an extension to the POSIX.4 nanosleep() function — a new function, actually — to wait until a specified time using the POSIX.4 high resolution timespec.

- **Barrier Synchronization objects**: Independently of these being proposed for POSIX.4, they were also spec'ed by POSIX.14 — the Multiprocessing Working Group. Because POSIX.14 is a Profiles Working Group, they need to have any new interfaces that they propose put into one of the System Interfaces Working Groups' drafts. The POSIX.14 group had already made tentative arrangements for a number of new synchronization primitives to go into POSIX.1a, so the POSIX.4 Working Group may drop this.

- **Enhancements to POSIX.4**: Yes, the ink is barely dry on the official standard approval, and we're thinking about "enhancing" POSIX.4. That's because some people have implemented, or are in the process of actually implementing it. The one extension presented was to POSIX.4 message queues to make registration for notification of message arrival, via *mq_notify()*, optionally persistent.

Other "housekeeping" items, such as resolution of conflicts or unintended ambiguities between POSIX.4 and Pthreads, may come up in time for a POSIX.4d effort.

The January working group is expected to me more of the same: POSIX.13 ballot resolution and new proposals. There are also coordination issues between the POSIX.4 and POSIX.20 — Ada binding to POSIX.4 — Working Groups, and with the Distributed Realtime group, POSIX.21 to be addressed as they arise.

Report on POSIX.7: System Administration

Matt Wicks <wicks@fnal.gov> and Keith Duval <duval.vnet.ibm.com> report on the October 18-22, 1993 meeting in Bethesda, MD:

POSIX.7 is divided into three separate groups, each producing their own standard:

- POSIX.7.1 - Printing Administration
- POSIX.7.2 - Software Installation and Management
- POSIX.7.3 - User and Group Administration

Of all the work of POSIX.7, the work of the printing group is most advanced, with the initial formal ballot conducted in June-July, 1993. The print standard is based on MIT's Palladium, which is a distributed print management system and also the base technology for OSF's offering in the print management arena.

The working group explicitly decided to reject using *lpr* or *lp* as the basis of the standard, believing that neither really addressed all of the issues of a distributed printing system.

The ballot was generally positive, so there seems to be some willingness within the standards com-

munity to approve System Administration based standards. It remains to be seen if both vendors and users are ready and willing to migrate to a totally new printing system.
Commencing the week of October 18th, the POSIX.7.1 Printing standards group met in Bethesda. A great deal of progress was made toward producing a final document which satisfies the overwhelming majority of interested parties, and while resolving the objections and comments is a daunting task, the committee was formed, and objectives and means for achievement of the goals at hand were delineated.

Moreover, there were a number of excellent suggestions from the balloters which will improve the overall standard and implementations derived from same. Further, it was readily recognized by all who participated in the arduous task of interpretation and response to the objections and comments in Bethesda, and by all who are credible in the field, that this represents a significant enhancement of the art with respect to distributed systems technology. Finally, it was generally agreed that 'times have changed' and, if we allow ourselves the intellectual stimulation, change is good for us. In the print technology area, it was increasingly obvious during the week that the 'old' is just a bit too old to be relevant any longer, other than as grist for whimsey and fond recollection of much simpler systems challenges and times.

The Software Management standard is based on Hewlett Packard's software installation package, with some contributions from the SVR4 software installation package. (The HP system is also the base technology for the software management portion of OSF's Distributed Management Environment.)

There are two primary goals of the standard. One goal is to provide a standardized command line interface for all of the typical software management tasks. These include commands to install and remove software, configure software, and list and verify software. This goal allows administrator portability since the software management process will work the same on different machine types.

The second goal is to define a standard software package layout. This goal allows media portability. Software packaged in the standard layout would be able to be managed by any POSIX conformant implementation.

For a good explanation of additional details of the standard, I recommend obtaining a copy of the proceedings of the most recent USENIX LISA conference. Barrie Archer provided a very good paper that not only explains the standard, but also some of the reasons why certain decisions were made.
I have been involved in the Software Group since its formation over two years ago. This meeting has a significantly different flavor as the group is nearing completion of its initial work, planning to go to ballot after the April, 1994 meeting. Although there were several heated discussions on several technical issues over the course of the week, in general the work was focussed on "fine tuning" the document.

The next two meetings will be dedicated almost exclusively to editorial issues and attempting to resolve any discrepancies between different sections in the document.

A separate snitch report is being written by a member of this working group. However, I did want to use this opportunity to encourage other people to get involved.

The User and Group Administration work is in its early stages and is being done primarily by two individuals (a third person joined them this week) both of whom are vendor representatives. Here is an opportunity to get involved and make a difference in the standards arena. Otherwise, you will have to accept what is produced by a very small group of people.

Participation in POSIX does take time, but it is well worth it. Send me mail if you would like more information on how to get involved.

Report on Automated Testing BOF

Kathleen Liburdy <liburdy@hubcap.clemson.edu> reports on the October 18-22, 1993 meeting in Bethesda, MD:

The fourth Automated Testing BOF met on Wednesday afternoon during the week of POSIX in Bethesda, MD. This group provides a forum for the discussion of alternative and progressive approaches to conformance testing. Announcements and discussion related to the group are posted to the mailing list <*oats@stdsbbs.ieee.org*>.

In the opening remarks, a Project Authorization Request (PAR) was announced for POSIX.5 (Ada) test methods. This project will explore the potential application of formal specifications and automated testing in POSIX test methods. In particular, the assertions will be developed using the Clemson Automated Testing System (CATS) assertion language. These assertions are English-like in nature and can be automatically translated into an executable test suite. The decision to

apply formal specifications in test methods development was strongly supported by the POSIX.5 working group. The PAR was approved by the Sponsor Executive Committee (SEC), and the first meeting for this effort is scheduled for January 1994 at the POSIX/Irvine meeting.

The first presentation was an update on the ADL Project by Shane McCarron. The ADL Project is a four year project sponsored by the Japanese Ministry of International Trade and Industry. The project is being managed by X/Open and the primary research is being conducted by Sun Microsystems Laboratories. The mission of this project is to improve the test suite creation process through automation.

Each version of each deliverable is being reviewed by a public review group (*XoPub-ADL@xopen.co.uk*). Several sets of draft documents have been submitted for public review, including version 0.5 of the ADL Language Reference Manual and ADL Translator Design Specification. The ADL Project Quality Plan has also been delivered, and is now complete at version 1.0. The next deliverable of the ADL Project is November 1, which includes ADLT Design Spec 1.0 Alpha, ADL Language Reference Manual 1.0 Alpha, and other related documents. All these documents will be placed on the uunet ftp site ftp.uu.net under the directory /vendor/adl.

A technical briefing by Alberto Savoia of Sun Microsystems on the ADL Project was announced for the following morning, and all AT BOF participants were invited to attend. Alberto agreed to present a technical update on the ADL Project and discuss related issues at the next AT BOF in Irvine, CA.

The next presentation was "Automated Testing of POSIX Subsets" by Jim Leathrum. As part of the continuing development of CATS, experiments have been undertaken to investigate the issues associated with specification and execution of tests for subsets of standard interfaces. As part of this work, the CATS test harness has been enhanced to allow the test developer to create subsets of both the specification and the implementation of the system under test. A version of the CATS facility with the subsetting capabilities and the corresponding user manual are scheduled for release in January.

The ability to subset specifications and implementations in the CATS environment has led to many new issues which could not be addressed before. Jim discussed issues such as subset integrity, testability and granularity which are currently being investigated with the CATS facility. Many of these issues were also raised by the POSIX Subset Ad Hoc group. At the conclusion of the presentation, Lowell Johnson asked about the possibility of applying CATS to the POSIX subset dilemma by implementing and experimenting with some of the proposed POSIX subsets. Jim agreed that this would be an interesting application for CATS and indicated that this idea would be considered in future work.

Roger Martin, chair of the Steering Committee for Conformance Testing (SCCT), expressed an interest in being responsive to issues raised in the AT BOF. He stated that the decision in April 1993 to rescind testing requirements should be viewed as an opportunity to explore new approaches to testing. He also announced an invitational workshop on alternative testing methodologies to be hosted by NIST. The precise date for this workshop has not been determined, but the general time frame is spring 1994. The purpose of the workshop is to bring together major players in the field of conformance testing and collectively identify ways to cooperate in the pursuit of improved testing capabilities.

The fourth issue of the OATS newsletter was distributed. In addition to articles related to the AT BOF presentations, the newsletter includes "CATS in the Classroom," "Software through Pictures: The T Tool," and "DejaGnu Product Description." Submissions for future issues are invited and should be sent to *<liburdy@hubcap.clemson.edu>*. Requests for issues of the newsletter may also be sent to this email address.

Report on Fault Management Study Group

Stephen Hinde <S.Hinde@frec.bull.fr> reports on the October 18-22, 1993 meeting in Bethesda, MD:

Do you know the difference between Fault Tolerance, High Availability, a fault, a failure and an error? If so you could consider joining the "Fault Management Study Group" at the next POSIX meeting at Irvine.

October was the first meeting of this group, following BOF sessions at the two previous POSIX meetings. The status of the group is a "Study Group" preparing a Project Authorization Request (PAR). The healthy participation would indicate that Fault Management is something organizations are interested in sending people to work on which is one of the basic criteria for success in these hard times.

A number of existing documents are being studied as base documents, including the "UNIX International High Availability Working Group report," which was contributed by UI at the previous BOF, and a draft of the "ANSI X3.T8-1993 Fault Isolation Information Characterization for Information Technology."

The group found itself with an awesome "laundry list" from the submitted requirements. The requirements ranged from a framework to improve application availability to a framework to improve platform availability. The required system scope is not limited, and ranges from single CPU systems to Symmetric Multiprocessor systems and clustered systems.

The group debated whether it was to be a new PAR, or a sub-PAR of an existing group. The word "Management" had led some to ask whether the group would end up as a sub-PAR of POSIX.7, the System Management group. However some of the objectives of the group were clearly outside the area of System Management, and a number of alternative titles for the group were being considered, for which the current favorite was "Services for Dependable Systems." The PAR/sub-PAR debate will be easily settled when the PAR submission and scope documents are complete.

The work of fleshing out a Fault Management Process Model largely dominated the meeting, this is a model that would allow the decomposition of the error detection and error treatment steps, and allow the identification of the APIs involved. The model was mapped against several implementations as a sanity check. The behavior and definition of the building blocks of the process were examined, including Error Detection, Symptom Encoding, Error Logging, Diagnosis, Notification, Reconfiguration and Recovery. Possible areas for standardization could include APIs for Error Logging, Error Reporting, APIs for recovery modules, and a "fingerprinting" technique for uniquely identifying faults.

Bradford Glad, of ISIS, gave a presentation of HORUS a distributed toolkit layer designed to build distributed fault tolerant systems. The key ideas include virtual synchrony, a fault tolerant membership service, process groups, and a reliable broadcast protocol. New work includes a high level abstraction called the Uniform Group interface.

This working group spent an intensive week looking at a wide range of topics in the fault tolerant arena. The acid test is going to be selecting a hit list of topics for standardization, ready for the PAR submission at Tahoe.

If you are interested in more information on the group why not contact the group chair Helmut Roth, *<hroth@relay.nswc.navy.mi.>*?

Computers Could be Like Autos

Humor by Dave Taber
<David.Taber@Eng.Sun.COM>

What driving your car would be like if operating systems ran it:

Windows: You'd get into your car and drive to the store very slowly because five boulders are dragged along behind the car.

Windows NT: You'd get into your car and write a letter that says, "Go to the store." Then you'd get out of the car and mail the letter. The dashboard of the car would glow knowingly.

OS/2: After fueling up with 60 gallons of kerosene, you'd get into the car and drive to the store with a motorcycle escort and marching band in procession. Halfway there, the car would catch fire.

Taligent: You'd walk to the store with Ricardo Montalban, who tells you how wonderful it will be when he can fly you to the store in his jet.

UNIX: You'd get in a diesel locomotive and start looking around for the "go" switch. The control panel has 150 unmarked levers. The speedometer calibrations start at 90 miles an hour, and go up from there.

The Bookworm

by Peter H. Salus
<peter@uunet.uu.net>

Some of the publishers were fairly unproductive between Thanksgiving and Christmas, so I've got less to wade through and write about. But Fortran, Frame Relay, and (yes, Virginia...) the Internet came through, as well as two non-computer items that I thought would be of interest.

Fortran

FORTRAN, in the version of Backus and a dozen others was the first computer language to which I was exposed, in 1957. Until I encountered Lisp 1.5 in the mid-1960s, I limped along with FORTRAN II. Years passed. I became an APL user. Then a UNIX/C user. Now a quintet of authors has brought out a volume on High Performance Fortran and Kerrigan has done one on Fortran 90.

Fortran 90, ANSI's successor to Fortran 77, has many features that weren't around 15 years ago (e.g., lots of new control structures, new data storage mechanisms). As there are zillions of programs in F77 and as there are many programmers who use Fortran, a good book on migrating to the most recent standard version was called for. And O'Reilly & Associates have supplied one. Though I mentioned ANSI above, because computer languages fall under the X3 committee rubric, Fortran 90 is now an international standard: ISO/IEC 1539:1991. This is a thorough, well-organized book; the sort of thing I've come to expect from ORA. James Kerrigan has done a really fine and useful job.

Koelbel, Loveman, et al., have provided us with a volume that builds on Fortran 90. In November 1991, DEC sponsored a BOF at Supercomputing '91 to talk about their High Performance Fortran project. The following January a High Performance Fortran Forum was convened. HPFF met about a dozen times, received input from X3J3, and brought forth a language specification in May 1993. The chapters in this book are based on that spec. The volume is intended as a genuine tutorial, and it is an excellent one. I doubt whether I'll ever use HPF, but if I have to, I'll turn to this first.

Frame Relay

As most of you probably know, Frame Relay is a communications protocol that's designed for bursty applications. It has developed out of ISDN because of the inadequacy of X.25. Philip Smith has turned out a workmanlike book that surveys the status and future of frame relay. I was not thrilled by the chapter on the standardization process, but this was primarily because I am interested in standards and am becoming a curmudgeon from associating with Stan Kelly-Bootle. Frame networks really do offer a better quality of service than do traditional networks. If you're in a place where you need to know at least something about frame relays, this book is better than others I've seen.

Net ball!

If you need a book for a true beginner on the Internet, buy Kehoe. One step up is Dern's weighty tome. For the veteran user, there's Krol. I've got all those and more. The two most recent ones are Hahn and Stout's *The Internet Complete Reference* and Cronin's *Doing Business on the Internet*. They are both big, heavy books.

I really like Hahn's *Student's Guide to UNIX*. This one is less to my taste. It's not a bad book, but it lacks something. Dern went out of his way to supply pointers, realizing that Internet lists are out-of-date by the time you print them out. Hahn and Stout have many pages of lists: Usenet groups, suppliers, and domains, for example. Several of the groups I read aren't there. Many suppliers aren't there either. The best chapters are the ones on Internet Relay Chat and Gopher, Veronica, and Jughead. Mary Cronin's book is a superficial presentation for vice presidents or other illiterati. It is chock full of errors and misprints. Cronin has a list of USENIX folks in her thank-yous. She could have spelled their names right. I guess every publisher needs to have an Internet book out. This one's for those who intend to stroll along the verge of the Information Highway, hoping for a stray hubcap.

The Literary Element

The University of California Press is publishing an edition of the Victorian author Thomas Carlyle in eight volumes. The first of these has just appeared. While I am actually re-reading "On heroes, hero-worship and the heroic in history," I don't expect most readers of this to do so. What I do think they should do, however, is contemplate the use of the UNIX Operating System. Murray Baumgarten of UCSC, the Editor-in-Chief of the Edition is relying quite heavily on UNIX. What he (delicately) calls "The application of electronic technology" has been used at every stage in the project, from the collection of evidence through to the final typesetting. As someone who sees the computer as the most valuable tool for a variety of chores, I was just thrilled to see this volume and to realize that there was now a machine-readable archive of Car-

lyle, purely as a by-product of the edition. This is a wonderful result of utilizing UNIX. I can't wait for "Sartor Resartus."

And the other arts

Emmer's *The Visual Mind: Art and Mathematics* is a beautiful book, full of interesting articles by mathematicians interested in the visual arts and visual artists involved in the mathematical aspects of their work. The biggest failing (if it is one) is that so few of the 35 chapters are more than half a dozen pages in length. I really wanted, for example, Harriet Brisson's piece on visualization and Anthony Phillips' on Roman Mosaic Mazes to go on for at least another dozen pages each. The same might be said for the two articles on Platonic solids. The black and white and color plates are splendid.

Charles H. Koelbel, David B. Loveman, Robert S. Schreiber, Guy L. Steele Jr., and Mary E. Zosel, *The High Performance Fortran Handbook*. MIT Press, 1994. 300pp. ISBN 0-262-61094-9. $24.95.

James F. Kerrigan, *Migrating to Fortran 90*. O'Reilly & Associates, 1993. 361pp. ISBN 1-56592-049-X. $27.95

Philip Smith, *Frame Relay: Principles and Applications*. Addison-Wesley, 1993. 268pp. ISBN 0-201-62400-1. $41.95.

Harley Hahn and Rick Stout, *The Internet Complete Reference*. Osborne/McGraw Hill, 1994. 818pp. ISBN 0-07-881980-6. $29.95.

Mary J. Cronin, *Doing Business on the Internet*. Van Nostrand Reinhold, 1994. 300pp. ISBN 0-442-01770-7.

Michael K. Goldberg, Joel J. Brattin and Mark Engel, eds., Thomas Carlyle, *On Heroes, Hero-Worship, and the Heroic in History*. University of California Press, 1993. ISBN 0-520-07515-3.

Michele Emmer, ed., *The Visual Mind: Art and Mathematics*. MIT Press, 1993. 274pp. ISBN 0-262-05048-X. $39.95.

Learning the Korn Shell

Learning the Korn Shell by Bill Rosenblatt. O'Reilly and Associates, 1993. 338 pages. ISBN 1-56592-054-6. Soft cover, $27.95.

Reviewed by Adam S. Moskowitz
<adamm@world.std.com>

It wasn't all that long ago that the Korn Shell was available only through AT&T's Toolchest. Folks who were willing to pay for and port source code had it, and the rest of the world suffered without command-line editing or used the arcane *csh* editing features. Now, the Korn Shell is found on every UNIX system and even on some DOS boxes. Even though the Korn Shell differs in some significant ways from the POSIX.2 shell, it is still, in my opinion, a better shell than (Bourne) *sh*, both for programming and especially for interactive use. This book from O'Reilly and Associates is a good book from which to learn how to use the Korn Shell.

The book is written for "casual" users, although there's information in here for more experienced folks as well. Keeping in mind the intended reader, Rosenblatt starts of with a brief overview of shells, files, wild-cards, I/O redirection, pipes, background jobs, and quoting. The second and third chapters cover command-line editing and customizing the environment.

Chapters four through eight contain the meat of the book. Here, Rosenblatt takes the approach of explaining the features of the Korn Shell from the perspective of the tasks a user is likely to want to perform; for the intended audience, this approach works and Rosenblatt does a good job of it. For many features, Rosenblatt starts with a brief discussion of the feature, then gives the reader a task to solve. The tasks are well thought-out, clearly described, and the application of the feature(s) just presented are clear. He then goes on to solve the task, step by step, showing portions of the solution along the way as well as the resulting output from the code fragments. Where appropriate, the full solution is then shown. If the solution uses a UNIX command, the command is briefly discussed just before it's used. While most of the examples are of tasks the average casual user might perform, or something similar enough that it's still a useful example, Rosenblatt provides a few advanced examples; one is an implementation of the well-known directory stack functions.

Chapter nine is devoted to a single advanced example: A debugger for Korn Shell scripts. This example introduces the user to signals, *exec*, *trap*, and makes heavy use of arrays and functions. In addition to being a good example to bring everything together, the debugger works and looks like it would do a reasonable job in real-life situations.

Rosenblatt devotes chapter ten to Korn Shell administration, including a brief discussion of system security and how the Korn Shell features of tracked aliases and privileged mode can be used to make set-uid shell scripts less of a problem. While this chapter is in no way comprehensive, it does point out a few things the casual systems administrator needs to know when installing the Korn Shell.

Appendix A discusses related shells and the future of the Korn Shell. Here, Rosenblatt gives a brief overview of six shells (*sh*, the POSIX.2 shell, *wksh*, *pdksh*, *bash*, and the MKS shell for DOS); in addition, he discusses the major differences between the Korn Shell and each of these other shells. This will no doubt be useful to folks porting shell scripts from other shells to the Korn Shell. This appendix also contains a list of the differences and new features found in the beta version of Dave Korn's latest efforts. Unfortunately for those of us eager to get our hands on that beta version, Rosenblatt notes that the "negotiations between AT&T and USL . . . could very well postpone the new shell's public release for a couple of years or more."

The second appendix is a reasonably complete set of reference lists. Invocation options, commands and keywords, options, test operators, and more, each with a brief description of function and, where appropriate, a chapter reference. The only thing lacking is a list of the Korn Shell operators such as [[, ##, %%, etc.

The book contains a small handful of errors of varying degrees of severity; most of them are font and spacing problems in code examples. While this is understandable, as code examples look like gibberish to most printers (the people, not the machines), in some cases it could cause confusion to neophyte *ksh* users. In one or two places, there are pathological cases of file names or directory structures that could cause the examples shown to fail. The rest of the errors may well be errors to my eyes only because I am overly fussy about grammar. Lest I sound too negative, these errors do not detract from the overall (high) quality of the book.

Overall, I thought the book was well written. The approach is a good one, especially for users and programmers with only a few years of experience. There's also enough information that even long-time shell hackers might learn something.

Unless you're a dyed-in-the-wool Korn Shell user looking for a concise reference book, I recommend buying this book.

Learning Perl

Learning Perl by Randal L. Schwartz. O'Reilly & Associates, Inc., 1993 ISBN: 1-56592-042-2. 246 pages. Paperback.

Reviewed by Rob Kolstad
<kolstad@bsdi.com>

The title of the book gives away all its secrets: *Learning Perl.*

This book is a far more gentle introduction to the powerful perl programming paradigm than the previously published reference work, which I think is too complex for a first-time perl user. It is even gentler, in many ways, than reading the man-page, which is the way I'm sure many people have started their perl programming practice.

For beginning perl users, it fills a niche previously filled only by tutorials taught be a small number of perl experts. The book's level of perl sophistication is fairly consistent and is a full step less complex than the companion reference or the man page. The style is breezy and far less cutesy than the reference book. I found the typesetting and general book layout to be remarkably easy on the eyes.

The first 34 pages comprise an appetizer. A script is evolved, growing in sophistication (by using ever more perl features). Perl's power is exposed in a way that could easily motivate a skeptic to continue reading the subsequent material in an effort to master its content.

The book's (negligible) fault lies in its limited scope. I tested the book's index by looking up the 'split' operator. Only the first two arguments were ever discussed. Likewise, the nuances of using 'eval' were not covered. This means that a student of this book can read and understand it completely and only get ten lines into some of the more complex scripts on the network and be unable to proceed without consulting the reference book or the man page.

Also, I would have ordered some of the paragraphs differently and would have concentrated a bit more on some sections than others (e.g., two pages on full array assignment precede two pages on array element assignment, which I would have put first). No big deal.

All-in-all, *Learning Perl* is a fine introductory text that can dramatically ease moving into the world of perl. The more aggressive reader will want to supplement the book with others references, of course. Nevertheless, the UNIX community (with its myriad of toolsets) too often lacks the kind of tutorial that this book offers.

Announcements/Calls for Upcoming Events

On the following pages are Announcements and Calls for Papers (CFPs) for upcoming USENIX events. Watch *comp.org.usenix* for regular postings of updates, student grant applications, registration, and hotel information.

GUUG 50
SANS III 51
C++ Tutorials 52
USENIX/SAGE Security & System Administration Management Seminars, at the UniForum Conference, San Francisco, CA 53
CFP: High-Speed Networking, Berkeley, CA 55
CFP: LISA 1994, San Diego, CA 57
CFP: Very High Level Languages (VHLL), Santa Fe, NM 59
CFP: Operating Systems Design and Implementation- **JUST ANNOUNCED** 61

Call for Papers

German UNIX USER GROUP: Annual Conference and Exhibition Wiesbaden, Germany September, 20-22, 1994

The GUUG Annual Conference and Exhibition is the largest exhibition and best attended conference in Europe devoted to technical, application and integration aspects of Open Systems, especially UNIX. It is going to reach about 10,000 engineers, programmers, technical managers and users of Open Systems.

The GUUG event will be centered around the conference with 3 parallel sessions: 2 tracks of technical and one track of product-oriented talks. These technical- and marketing-oriented talks are complemented by reports about user solutions based on UNIX or another open systems approach.

Topics of main interest:

Technology
- Object oriented methods and tools, reusability
- UNIX in client-server architectures
- UNIX on PC vs. Windows/NT
- Multimedia
- OLTP
- System and Network management

Application
- Office communication with UNIX
- Commercial applications under UNIX
- UNIX in manufacturing
- Migration from mainframes to client-server architectures

Interested authors are invited to send an abstract of about 2 pages (no slides) outlining the basic ideas of their intended talk including name, company/institution, address, phone number until February 14, 1994 to:
NETWORK GmbH
Wilhelm-Suhr-Strasse 15
D-31558 Hagenburg
Tel. (05033) 7057; Fax (05033) 7944

All abstracts will be reviewed carefully with respect to quality, ingenuity, and technical relevance of the paper. Speakers will be chosen accordingly.

The authors will be informed about acceptance until April, 19. The full papers should be sent in till July 6 in order to be included into the conference proceedings.

GUUG: German UNIX User Group
Email: *guug@guug.de* phone: +49 89 5707697
Programme Chair: Ms. Ulrike Weng-Beckmann
Email: *weng-beckmann.muc@sni.de*
Phone: +49 89 636 3030

ATTENTION! SYSTEM ADMINISTRATORS

THE THIRD ANNUAL SYSTEM ADMINISTRATION, NETWORKING AND SECURITY CONFERENCE

SANS III

April 4-8, 1994
Stouffers
Concourse Hotel
Washington, D.C.
(Arlington, VA)
(At Cherry Blossom Time)

Presented By OSCB, In Cooperation With The USENIX Special Technical Group SAGE SAGE

PRELIMINARY PROGRAM

FOR CONFERENCE INFORMATION, CALL 719-599-4303

PROGRAM COMMITTEE

Rob Kolstad
(Chairman) BSDI

Paul Mockapetris
(SAGE Liaison) cisco

Bill Howell
(Management)
Univ. of North Carolina

Tom Christiansen
(USENIX Liaison)

Matt Bishop (Security)
Univ. California at Davis

E. Scott Menter
(Commercial UNIX)
Enterprise Systems Management

Alan Paller
(Downsizing To Open Systems)
Computer Associates

COURSES

Monday-Wednesday, April 4-6, 1994

These practical courses bring together, for the first time, seven of the top-rated tutorial instructors in system administration and security to teach the most requested and useful topics in this area. And, for the first time at SANS, you'll be able to switch among the Tuesday tutorials, picking half-day segments most valuable to you.

MONDAY, APRIL 4

Key Security Challenges and Solutions: Part I
Matt Bishop

Introduction to the Tools and Philosophy of System Administration
Rob Kolstad

TUESDAY, APRIL 5

Key Security Challenges and Solutions: Part II
Matt Bishop

SendMail Administration and DNS Management
Rob Kolstad and Tina Darmohray

Practical Perl Programming
Tom Christiansen

Key Topics In Network Management for Administrators
Evi Nemeth and Trent Hein

WEDNESDAY, APRIL 6

Firewalls and Other UNIX Security Issues and Tools
Rob Kolstad and Tina Darmohray

Managing Managers, Staff, Users and Systems
Bill Howell

Key Topics In System Administration
TBA

TECHNICAL CONFERENCE

Wednesday-Friday, April 6-8, 1994

INVITED PRESENTATIONS

Keynote:
Computer Crime: Are You At Risk?

The Human Side Of System and Security Administration
- Managing Your Manager And Your Tasks
- Salaries And Workloads
- Win-Win Interaction With Users
- System Adminstrator Ethics: A Large Group Discussion

Tools For System Administration
- The Best Of The Free Tools From The Net
- Popular and Effective Third-Party Tools
- Applications of Perl
- Guided Tour of the man Pages
- Backups
- Internet Discovery and Retrieval Tools

Tools For Computer Security
- How To Identify Security Holes
- Security Perspectives Panel With Government, Commercial, And Academic Views
- Problems Of Interoperation In Multi-Vendor Installations
- Responding To Intruder Incidents
- Breaking Into Banks

Managing Downsized Commercial Systems
- Three Paths To Client Server Computing
- Managing UNIX Computers Using IBM Mainframe Operations Staff
- The Dark Side of Distributed Management
- Is The Mainframe Dead? Dispelling The Myth
- Large Commercial UNIX Sites: Directors Discuss The Issues

Panel On Future Of System Management

Peer-reviewed papers are being evaluated at press time.

COMMERCIAL TOOLS TRACK: A SANS Exclusive!

Get hands-on access to the leading third-party administration and security management software packages with substantive briefings on each of them.

SPECIAL TRACK: The Guru Is In!

Facing Any Challenges? Have a particular problem? Find answers and solutions to your problems by talking one-on-one with the experts. Guru sessions will be scheduled throughout the Technical Conference.

Birds Of A Feather Sessions

Important: Absolute limit of 250 participants at SANS III. Please call or email early for registration information.

For Complete Program and Registration Information
PLEASE CONTACT THE CONFERENCE OFFICE
8902 Edgefield Drive, Colorado Springs, CO 80920, Phone: 1-719-599-4303, FAX: 1-719-599-4395, Email: sans@fedunix.org

C++ Tutorials

C++ Conference
April 11-14, 1994
Cambridge, MA

Tutorial Program
Monday and Tuesday, April 11-12

Designing and Implementing Effective Classes
Scott Meyers, Software Development Consultant

Intended Audience:

Programmers and managers involved in the design and implementation of C++ classes for real products. Participants should already know C++, but expertise is not required. People who learned C++ through an earlier tutorial, as well as people who have been programming in C++ for some time, should come away from this tutorial with useful, practical information.

Object-Oriented Network Programming with C++
Douglas C. Schmidt, University of California, Irvine

Intended Audience:

The tutorial is intended for developers who are familiar with general object-oriented design and programming techniques (such as modularity and information hiding), fundamental C++ programming language features (such as inheritance, dynamic binding, and parameterized types), basic systems programming concepts (such as process/thread management, synchronization, and interprocess communication), and networking terminology (such as client/server architectures and TCP/IP). The purpose of the tutorial is to illustrate by example how OO and C++ significantly simplify and enhance network programming. Portions of the tutorial material examine C++ source code to illustrate key points in the examples.

Design Patterns – Elements of Reusable Object-Oriented Software
Richard Helm, DMR Group and John Vlissides, IBM T.J. Watson Research Center

Intended Audience:

Architects, system designers and programmers who are interested in the design of flexible and reusable object-oriented software. Participants should have had experience in object-oriented design and have a working knowledge of object-oriented concepts such as polymorphism, types and interface inheritance, and how they are realized in C++.

Templates, Containers, and Iterators
Andrew Koenig and Rob Murray, AT&T Bell Laboratories

Intended audience: C++ programmers who are want to learn how to use templates. A basic knowledge of C++ is assumed; no templates knowledge or experience is necessary.

Templates are among the most important recent developments in the C++ language. Templates provide a way to represent a family of function or class definitions that differ only in the types of the things they use.

The most common use of templates is to define "container classes:" classes that contain objects of other, user-specified classes. Typical container classes include sets, lists, associative arrays, and other such data structures.

A C++ Programmer's View of CORBA
Steve Vinoski, Hewlett-Packard

Intended Audience:

C++ programmers interested in distributed object computing based on the Common Object Request Broker Architecture (CORBA) specification of the Object Management Group (OMG). No knowledge of CORBA is required, though some knowledge of basic distributed computing concepts might prove helpful. A good working knowledge of the C++ language is assumed.

A Taste of Fresco
Mark Linton, Silicon Graphics

Intended Audience:

This tutorial is for C++ developers of interactive applications or those interested in studying graphical user interfaces as a case study for the design and implementation of a class library. Attendees should have experience programming in C++ or at a minimum be familiar with basic object-oriented programming concepts. Familiarity with graphical user interfaces will be helpful.

The full program will be mailed to members in early February. For more information, please contact the USENIX conference office via phone: 714/588-8649, FAX: 714/588-9706, or email:<*conference@usenix.org*>.

SECURITY & SYSTEM ADMINISTRATION MANAGEMENT SEMINARS

at the UniForum Conference

MARCH 21 & 22, 1994

MOSCONE CONVENTION CENTER
SAN FRANCISCO, CALIFORNIA

Sponsored by

USENIX and SAGE THE SYSTEM ADMINISTRATORS GUILD

ORGANIZING COMMITTEE:

Paul Evans, SRI International, Ellie Young, USENIX, and Daniel Klein, USENIX

TO REGISTER:

You may choose to attend one or both days of either of the two seminars: One day $295; two days: $500 (before February 18; fee includes lunch.) Call 1-800-225-4698 (in U.S.A.) or 1-508-879-6700 to request UniForum '94 registration form.

WHO SHOULD ATTEND?

UNIX system managers and system administrators charged with maintaining security or with meeting their organization's system administration needs. You may choose to attend one or both days of either of the two seminars. The 1994 UniForum Conference is the first to offer seminars sponsored by the USENIX Association and SAGE, the System Administrators Guild.

These intensive seminars explore effective management techniques in two critical areas: protecting security and managing computing services. The first of the seminars examines immediate and long-term policies with which to combat security threats to your computing environment and, specifically, to the X-Window System. The second seminar provides insights and policies to achieve effective computing services management, while building a crack team of system administrators.

Seminar 1: SECURITY

If you are charged with protecting your organization's systems and networks against intrusion by malicious or mischievous parties (from both within and without), this session is for you. Learn how to recognize fundamental security risks. You'll also get an overview of the security features (and failings) available under UNIX, and the implications of security measures for interoperability and user-friendliness.

MONDAY • MARCH 21, 1994 • 9:00 AM–5:00 PM

UNIX System and Network Security

Rob Kolstad and Bill Rieken

The first half of the session covers issues such as file and directory protection, how to secure modems and multiple types of networks (LANs and WANs). You'll also learn about state-of-the-art security techniques such as Kerberos systems and network firewalls. In the afternoon, the seminar turns to the specifics of maintaining security in a UNIX environment. System accounting commands, the crypt command, passwords, file protections, and the mount command are all demonstrated in the context of system security. A custom security audit daemon is shown, along with sample output, to help you monitor your system. This half of the seminar also covers some details of UUCP security, SUID programming, Trojan Horses, and security audit and logging.

TUESDAY • MARCH 22, 1994 • 9:00 AM–5:00 PM

Security and the X Window System

Jeremy Epstein and Rita Pascale

What are the risks to security, and available solutions, posed by the X Window System? This seminar provides a basic introduction to both X and security – no advance in depth knowledge is needed. As the X Window System increases in popularity, so does concern about its security. Because X is an open, resource-sharing system, security measures are not easily retrofitted without damaging interoperability, and there are no quick and simple answers to security risks. Learn about new security-enhanced X systems being introduced by some vendors, and their implications for programmers and other users. Other topics covered include threats, current technologies, authentication, access controls, auditing, privilege, and denial of service. Use of authentication mechanisms is described in detail, including *xhost*, MIT magic cookies, Sun's Secure RPC, and Kerberos. Vendor-specific extensions to X for access control and privilege are presented. Alternate architectures are described for multi-level secure X-Systems.

Seminar 2: MANAGING AND DECISION-MAKING FOR THE SYSTEM ADMINISTRATION PROCESS

This two day, in-depth seminar teaches the how-tos of keeping computer systems on-line, of providing efficient computer services, and of building a crack team of system administrators. The format is a series of one- to two-hour long presentations in which veteran computing services managers share their experience. These presentations are designed to enhance your success as a manager of computing services and of a system administration staff.

MONDAY • MARCH 21, 1994 • 9:00 AM–5:00 PM

System Administrator Hiring and Job Performance

Tina Darmohray

Meet the challenges of building a highly effective system administration team at your site. As a manager of a growing staff of system administrators, this session provides you with a sophisticated understanding of what to look for when hiring, what is a good training program for your site, and, then, what kind of job performance you should expect.

System Administrators from the Manager's Perspective

William E. Howell

System administration is an exciting, but rapidly changing and ill-defined field. Learn to cope with the unique challenges of system administration management such as: growing out-of-touch with new technology; managing people who are doing work you've never done; and dealing with a poorly-planned management structure.

Managing for Ethical and Legal Computer Use

Rob Kolstad

Dealing with large user communities leads to new problems in data collection, software licensing, security, and ethics. We review the pros and cons of dozens of policies on issues ranging from e-mail privacy to use of copyrighted software. Get experience-based, common-sensical approaches to creating an organization-wide policy for computer use in your environment.

Making Sure Free Software is a Bargain

Rob Kolstad

Network software repositories contain a wealth of interesting and valuable software, such as very high level languages, compilers, source control systems, network protocols – available for no charge. In this section, we discuss a number of the free or public domain packages at length and show you how to find specific software using one of the many archival servers on the net. We also discuss the caveats of using this type of software and explain some of the drawbacks of various policies that may come with it.

TUESDAY • MARCH 22, 1994 • 9:00 AM–5:00 PM

Preventing Backup Disasters

Elizabeth Zwicky

No collection of system administration disasters is complete without a "no backup" disaster story. This session helps you avoid personal experience of what can be a true tragedy. The backup system that is well suited to one site may be completely inappropriate to another. This presentation discusses the issues involved in selecting a backup system and setting up a backup schedule uniquely suited to your site. We offer specific guidelines for choosing among the many available backup systems.

Making Do with a Limited System Administration Staff

Paul Evans

The industry trend to distributed computing is producing sites where the machine-to-system administrator ratio is frequently greater than 100 to 1. Learn how to live with this ratio by seeking strong management backing, using creative administration management, and, understanding where to focus system administration efforts. Learn techniques for implementing the time-saving software and hardware solutions available to tackle the special problems of running a large site with limited staff.

What Do System Administrators Really Do?

Elizabeth Zwicky

Managers, vendors, and system administrator are all frequently frustrated by what seems to be a massive communications gap. System administrators are often violently opposed to vendor's best-intentioned attempts to provide them with tools. They may be engaged in mutually puzzling dialogues with management on topics like staffing, appropriate goals, and performance measurements. This section discusses the causes and symptoms of this situation. We use a "day in the life" of a typical system administrator to contrast what system administrators really do with common management/user/vendor (mis)perceptions.

The Canons of Computer Services Management

William E. Howell

This section explores a variety of management approaches and styles and shares new ideas and views that will improve your management effectiveness. We consider your spheres of influence as a manager: your management, the computing environment you maintain, your vendors, your employees, your user community, and your organization's projects. For each of these we present canons or key principles that will help you be a stronger, more effective and more efficient manager.

Recognized Experts Sharing Their Experience

Dr. Rob Kolstad is Program Manager for Berkeley Software Design, Inc. Previously, he was staff engineer at Sun Microsystems' Rocky Mountain Technology Center in Colorado Springs, Colorado where he led development of Sun's new Backup Copilot product. Rob is editor of *;login:*, the newsletter of the USENIX Association.

Bill Rieken has taught at the University of Wisconsin and Southern Illinois University. A principal founder of .sh Consulting in Santa Clara, he publishes seventeen UNIX training books for the courses he teaches, and is co-author of *"Adventures in UNIX Network Applications Programming"* (Wiley, 1992)

Rita Pascale is a researcher in highly trusted distributed systems at ORA Corporation. Until recently, she worked on trusted windowing systems at TRW. She has written four papers on X and security.

Jeremy Epstein is Chief Engineer at Cordant, responsible for design and development of the trusted Novell NetWare product. Previously, he was a researcher in highly trusted windowing systems at TRW, and has written nine papers on the subject of X and security.

Elizabeth D. Zwicky is a Senior System Administrator at SRI International and a board member of SAGE. She has been involved in system administration since a Sun 2 was an exciting computer and 100 machines was a big network, and has published or presented dozens of papers on system administration topics.

Paul Evans is currently a Senior System Administrator at SRI International, where he and his colleagues manage a network of about 200 workstations on several dozen networks. He spent three years at MasPar Computer Corporation, where he singlehandedly ran a network of 150 workstations. He is a founding member of BayLISA and SAGE, and has served on the board of directors of both organizations.

Tina Darmohray is the Lead for the UNIX System Administration Team at Lawrence Livermore National Laboratory, where her group has responsibility for over 1,000 machines. In 1990, she installed the first firewall at LLNL and has since consulted with a number of sites in the Bay Area.

William E. (Bill) Howell is the Director of Computer Services for the Graduate Department of Computer Science at The University of North Carolina at Chapel Hill. In Bill's decade at UNC, he has overseen the expansion of the facility to more than 450 systems, 240 gigabytes of disk space, and a comprehensive high speed network for video, data, and voice. An experienced, college-level instructor, Bill also consults on project management, computer security, and management of computer service organizations.

1994 USENIX Symposium on High-Speed Networking

AUGUST 1–2, 1994
CLAREMONT HOTEL & RESORT
OAKLAND, CALIFORNIA

Refereed Paper Submissions

- Extended abstracts due: May 2, 1994
- Notification to authors: May 16, 1994
- Camera-ready final papers due: June 20, 1994

Symposium Schedule

Registration Materials Available:
- June 1994

Monday, August 1
- Keynote address, followed by technical sessions
- Reception
- Birds-of-a-Feather sessions

Tuesday, August 2
- Technical sessions

Field Trip

Wednesday, August 3
Join us for visits to two high-speed networking testbeds, XUNET/ BLANCA and CalREN, in Berkeley.

High-speed, high-capacity networks promise to change profoundly the way we compute. Fast, wide-area networks pose fresh challenges even for mature operating systems, such as UNIX. How will these innovations shape the design of future operating systems? Can we devise applications that fully (and productively) consume the bandwidth at our disposal?

The goals of this symposium are to encourage the UNIX and high-speed networking communities to commingle, to examine the issues and trends in high-speed networking, and to explore the impact of high-speed networks on systems and applications design.

The single-track symposium offers two days of technical presentations (followed by a field trip on the third day). Formally reviewed papers will be presented and published in the *Symposium Proceedings*. A copy of the *Proceedings* will be distributed to all attendees; additional copies may be purchased from the USENIX Association.

Symposium Topics

We seek presentations of original work on these (and related) topics:
- Network architectures
- Operating system support for high-speed networks
- Protocols
- Performance
- Network management
- Applications
- Practical experiences

Refereed Paper Submissions

If you are interested in presenting your work at the symposium, please submit an extended abstract as described below. The extended abstract should represent the final paper in "short form." Its object is to persuade the Program Committee that you will deliver a good 20–25 minute presentation and final paper.

The Committee needs to know that authors:
- are tackling a significant problem.
- are familiar with the current literature about the problem.
- have devised an original solution.
- have implemented the solution and, if appropriate, have characterized its performance.
- have drawn appropriate conclusions about what they have learned and why it is important.

Note that the Program Committee considers it unethical to submit the same paper simultaneously to more than one conference or publication, or to submit a paper that has been or will be published elsewhere, without disclosing this information with the submission.

If your paper is accepted, you are expected to provide a full paper in camera-ready form for publication in the *Proceedings* and to present your work at the Symposium.

How to Submit

A typical extended abstract is roughly 2500 words (5 pages). Indicate clearly whether the paper represents a design, an implementation or a system that is in wide use. You are encouraged to include references. Supporting material may be in note or outline form. If you wish, you may supplement the extended abstract with a copy of a full paper.

Please submit one copy of an extended abstract using *at least two* of the following methods:

- E-mail (*preferred method*) to: *net94papers@usenix.org*
- Mail to:
  Pat Parseghian, Program Chair
  AT&T Bell Laboratories, Room 2C-472
  600 Mountain Avenue
  PO Box 636
  Murray Hill NJ USA 07974-0636
- FAX to: Pat Parseghian +1 (908) 582-5857

Please, with your submission, include the following information about the author(s):

- Name (indicate which author will serve as the contact)
- Affiliation
- Daytime telephone
- Postal address
- E-mail address
- FAX number

For More Program Information

Refer questions about refereed paper submissions and other program concerns to the Program Chair:

- Pat Parseghian
  Telephone: +1 (908) 582-4229
  E-mail: *pep@research.att.com*

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.

Dates for Refereed Paper Submissions

Extended abstracts due:
May 2, 1994

- Notification to authors:
  May 16, 1994
- Camera-ready final papers due: June 20, 1994

Program Committee

- **Program Chair:**
  Pat Parseghian,
  AT&T Bell Laboratories

Bill Johnston, *Lawrence Berkeley Laboratory*

Tom Lyon, *Sun Microsystems*

Jeffrey Mogul,
Digital Equipment Corporation, Western Research Laboratory

Gerald Neufeld, *University of British Columbia*

Lixia Zhang, *Xerox PARC*

For Registration Information

Materials containing full details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be available early June 1994. If you wish to receive the registration materials, please contact:

- USENIX Conference Office
  22672 Lambert St, Suite 613
  Lake Forest, CA USA 92630
  Phone: +1 (714) 588-8649
  FAX: +1 (714) 588-9706
  E-mail:
  conference@usenix.org

8th USENIX Systems Administration Conference (LISA VIII)

SEPTEMBER 19–23, 1994
TOWN AND COUNTRY HOTEL
SAN DIEGO, CALIFORNIA

Co-sponsored by SAGE, the System Administrators Guild

The annual USENIX Systems Administration Conference provides a forum in which system administrators meet to share ideas and experiences. A growing success for the previous seven years, the USENIX Systems Administration Conference is the only conference which focuses specifically on the needs of system administrators. Its scope includes system administrators from sites of all sizes and configurations.

"Automation: Managing the Computer of the 90's" is the theme of this year's conference. The conference will focus on tools to help system administrators automate administration tasks and troubleshoot problems.

Tutorial Program

♦ **Monday and Tuesday, September 19–20, 1994**

The two-day tutorial program at the conference offers multiple tracks, with a total of as many as twelve half-day tutorials. Attendees may move between tracks, choosing the sections of most interest to them. Tutorials offer expert instruction in areas of interest to system administrators, novice through experienced. Topics are expected to include Networking, Advanced System Administration Tools, Solaris & BSD Administration, Perl Programming, System Security, and more.

Technical Sessions

♦ **Wednesday through Friday, September 21–23, 1994**

The three days of technical sessions program will include refereed paper presentations, invited talks, panels, Works-In-Progress (WIP) reports, and Birds-Of-a-Feather (BOF) sessions. The first track is dedicated to presentations of referred technical papers. Although papers of a traditional technical content are very welcome, the Program Committee is especially seeking papers on areas such as useful tools or solutions to system administration problems. Papers which are tutorial in nature would also be appropriate. The second track of the Technical Sessions will offer invited talks, panels, mini-workshops, and similar presentations, and we seek proposals for these presentation formats as well.

Conference Proceedings, containing all refereed papers and materials from invited talks and workshops, will be distributed to conference attendees. The *Conference Proceedings* will also be available from the USENIX Association following the conference.

Vendor Display

♦ **Wednesday, September 21, 1994, 3:00 pm–9:00 pm**

Well informed vendor representatives will demonstrate products and services useful to systems and network administration at the informal table-top display accompanying the USENIX Systems Administration Conference. If your company would like to participate, please contact Peter Mui at (510) 528-8649; FAX (510) 548-8649; E-mail: *pmui@usenix.org*

Conference Topics

The Program Committee invites you to submit to the refereed paper track of the technical sessions, as well as submit informal proposals, ideas, or suggestions for the various presentation formats of the second track, on any of the following or related topics:

- Automating Administration Tasks
- Distributed System Administration

Important Dates

Refereed Paper Submissions:

- Extended Abstract Submission Deadline: May 23, 1994
- Notification to Authors: June 24, 1994
- Final Papers Receipt Deadline: August 1, 1994

Registration Materials Available:

- July, 1994

- Problem Tracking
- Predicting problems before they happen
- System Administration standards
- Differences in OSF, Solaris, and ?
- Case studies – "This is the problem we solved and how we solved it."
- Career paths for system administrators ("Is there life after support?")
- Applications using emerging technology (C++, AI, etc.)
- Performance Monitoring
- Hardware-related topics: all about memory, installing disk drives
- Tools – Useful programs or solutions you have developed and wish to share

Refereed Paper Submissions

We strongly urge you to request a sample extended abstract by sending e-mail to *sample-abstract@usenix.org* or telephoning +1 (510) 528-8649.

The Program Committee requires that an extended abstract be submitted for the paper selection process. (Full-papers are not acceptable for this stage; if you send a full paper, you must also include an extended abstract for evaluation.) Your extended abstract should consist of a traditional abstract which summarizes the content/ideas of the entire paper, followed by a skeletal outline of the full paper.

Submissions will be judged on the following criteria: relevancy of topic, quality of work, and quality of the written submission.

Note that the USENIX conference, like most conferences and journals, considers it unethical to submit the same paper simultaneously to more than one conference or publication or to submit a paper that has been or will be published elsewhere.

Authors of an accepted paper will present their paper at the conference and provide a final paper for publication in the *Conference Proceedings*. Final papers are limited to 20 pages, including diagrams, figures and appendix and must be in troff or ASCII format. We will supply you with instructions and troff macros. Papers should include a brief description of the site (if applicable).

Where to Send Submissions

For submission to the refereed paper track, please send submissions by *at least two* of the following methods:

- (*Preferred method*) electronic (nroff/troff or ASCII) submission of the extended abstract; e-mail to: *dinah@usenix.org*
- FAX to the USENIX Association +1 (510) 548-5738
- Mail to: LISA 8 Conference, USENIX Association, 2560 Ninth St., Suite 215, Berkeley, CA USA 94710

For submission of all proposals other than extended abstracts of refereed papers, and for inquiries regarding the content of the conference program, contact the Program Chair: Dinah McNutt, Tivoli Systems, P.O. Box 202253, Austin, TX USA 78720-2253, +1 (512) 267-9381, E-mail: *dinah@usenix.org*

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.

Dates For Refereed Paper Submissions

- Extended Abstract Submission Deadline: May 23, 1994
- Notification to Authors: June 24, 1994
- Final Papers Receipt Deadline: August 1, 1994

Program Committee

- **Program Chair:** Dinah McNutt, *Tivoli Systems*

Tom Christiansen, *Consultant*

Trent Hein, *XOR Network Engineering*

William (Bill) LeFebvre, *Northwestern University*

Pat Parseghian, *AT&T Bell Laboratories*

Hal Stern, *Sun Microsystems*

Jeff Tate, *Bank of America*

Mark Verber, *Xerox PARC*

Neil Todd, *GID Ltd*

For Registration Information

Materials containing all details of the symposium program, symposium registration fees and forms, and hotel discount and reservation information will be mailed and posted to the net beginning July 1994. If you wish to receive registration materials, please contact:

- USENIX Conference Office
  22672 Lambert Street, Suite 613
  Lake Forest, CA USA 92630
  +1 (714) 588-8649
  FAX: +1 (714) 588-9706
  E-mail: *conference@usenix.org*

VHLL

USENIX Symposium on Very High Level Languages

OCTOBER 26–28, 1994
EL DORADO HOTEL
SANTA FE, NEW MEXICO

Important Dates

DATES FOR REFEREED PAPER SUBMISSIONS:

- Extended Abstracts Due: June 30, 1994
- Notifications to Authors: July 27, 1994
- Final Papers Due: Sept. 12, 1994

REGISTRATION MATERIALS AVAILABLE:

- August, 1994

Using very high level languages (VHLLs), programmers can assemble entire applications from large building blocks in just a small fraction of the time required if conventional programming strategies were used. These languages allow programmers to take advantage of increasingly available hardware cycles, trading cheap machine time for costly programmer time. Thus, VHLLs offer one of the most promising approaches toward radically improving programmer productivity.

UNIX has long supported very high level languages; consider *awk* and the various shells. Often programmers create what are essentially new little languages whenever a problem appears of sufficient complexity to merit a higher level programming interface – consider *sendmail.cf*. In recent years many UNIX programmers have been turning to VHLLs for both rapid prototypes and complete applications. They take advantage of these languages' higher level of abstraction to complete projects more rapidly and more easily than they could have using lower-level languages.

Some VHLLs such as *TCL*, *Perl*, *Icon*, and *REXX* have gained widespread use and popularity. Many others never see the public light. Some of these languages are special purpose, addressing a limited-problem domain (such as graphics, text processing, or mathematical modeling) using powerful primitives created for that specific problem. Other VHLLs are more general purpose in nature, but still much higher level than most traditional compiled languages. Some are stand-alone languages, while others are designed to be embedded in other programs. Many are interpreted, although some are compiled to native machine code; a few occupy a gap between both worlds.

Symposium Scope and Format

The USENIX Symposium on Very High Level Languages will spotlight these languages and their usefulness in leveraging certain kinds of tasks. The Symposium will introduce participants to concepts and approaches they haven't examined yet, and publish original work in these areas. Programmers will learn about the relative strengths and weaknesses and extract the key concepts that run through the various languages presented.

The USENIX Symposium on Very High Level Languages will run three days:

- Wednesday, October 26, will feature hour-long overviews by invited speakers of some of the more popular VHLLs in use today, such as *TCL*, *Perl*, *Icon*, and *REXX*.
- Thursday and Friday, October 27–28, will consist of refereed papers, tutorial-style invited talks on related topics, and panel discussions.
- Birds-of-a-Feather sessions will be held Wednesday and Thursday evenings, and a Reception will be held Thursday evening.

Papers on brand-new languages, on existing languages about which little or nothing has been published, on applications that use these languages in creative fashions not yet seen, and on experiences at extending existing languages (for example, adding windowing capabilities to *awk*) are all welcome. Papers should address designing, building, testing, debugging, and measuring the performance and usability of these languages, as well as reference and compare related work in the area. Mention both advantages and disadvantages of the approach selected. For applications using these languages, compare and contrast the design, development, and support effort that were required with this approach versus one using a lower-level language. Good papers will be of interest to people who use other VHLLs than the one described in the paper. For example, a paper describing a system built in a particular language will be much more interesting if it highlights some important feature of the language or problems with the language, or some issue relevant to VHLLs in general.

(conitnued on reverse side)

Program Committee

- **PROGRAM CHAIR:**
  Tom Christiansen, *Consultant*

Stephen C. Johnson,
Melismatic Software

Brian Kernighan,
AT&T Bell Laboratories

John Ousterhout,
University of California, Berkeley

Henry Spencer,
University of Toronto

Dates for Refereed Paper Submissions

- Extended Abstracts Due:
  June 30, 1994
- Notifications to Authors:
  July 27, 1994
- Final Papers Due:
  September 12, 1994

How to Submit to the Symposium

Persons interested in participating in panel discussions or organizing Birds-of-a-Feather sessions should contact the program chair as indicated below.

Submissions of papers to be presented at the Symposium and published in the *Symposium Proceedings* must be in the form of an extended abstract. The extended abstract should be 1500–2500 words (3–5 pages) and must be received by June 30, 1994. (If you do send a full paper, you must also include an extended abstract for evaluation.) The extended abstract should represent your paper in short form. Its purpose is to convince the program committee that a good paper and presentation will result. You should show that you are addressing an interesting problem, have surveyed existing solutions, have devised an innovative, and original solution, and have drawn appropriate conclusions about what has been learned.

All submissions should indicate the electronic mail address and telephone number of a principal contact. Authors will be notified of acceptance by July 27, 1994, and will be provided with guidelines for preparing camera-ready copy of the final paper. The final paper must be received no later than September 12, 1994. Note that the USENIX conference, like most confcrences and joumals, considers it unethical to submit the same paper simultaneously to more than one conference or publication or to submit a paper that has been or will be published elsewhere.

Please submit your extended abstracts to the program chair as follows:

EMAILED SUBMISSIONS (PREFERRED):

- must be in ASCII, troff (with the -me macro set or raw troff preferred), or Postscript form
- send to tchrist@usenix.org

HARD COPY SUBMISSIONS:

- via FAX to +1 (303) 442-7177 (Please refer to Tom Christiansen)
- via postal mail, please submit 6 paper copies to:

  Tom Christiansen
  USENIX VHLL Symposium
  2227 Canyon Blvd., #262
  Boulder, CO 80302

For Program and Registration Information

Materials containing full details of the symposium program, registration fees and forms, and hotel discount and reservation information will be mailed and posted to the net in August 1994. If you wish to receive thesc materials, please contact:

USENIX Conference Office
22672 Lambert Street, Suite 613
Lake Forest, CA USA 92630
+1 (714) 588-8649; FAX: +1 (714) 588-9706
Internet: conference@usenix.org

USENIX, the UNIX and Advanced Computing Systems Professional and Technical Association.

USENIX Symposium on Operating Systems Design and Implementation (OSDI)

NOVEMBER 14–18, 1994
MONTEREY, CALIFORNIA

Important Dates

Refereed Paper Submissions:

- Extended abstracts due: June 21, 1994
- Notification to authors: August 5, 1994
- Full papers due for editorial review: September 6, 1994
- Camera-ready, final papers due: October 4, 1994

Registration Materials Available: August 1994

USENIX®
The UNIX and Advanced Computing Systems Professional and Technical Association

Co-Sponsored (pending) by ACM SIGOPS and IEEE TCOS

The first OSDI Symposium will emphasize *innovative research* and *quantified experience and results* in operating systems. We seek papers describing original research concerning the design, development, implementation, and use of modern operating systems.

Background

The USENIX Association has previously sponsored three series of symposia on modern operating systems: Mach; Microkernels and Other Kernel Architectures; and Experiences with Distributed and Multiprocessor Systems. To eliminate overlap these three have been combined into a new, stronger symposium, OSDI. Although papers in the ancestral areas are emphatically solicited, we do not favor any particular OS architecture. In general, OSDI has a broader charter than its predecessors. The intent, however, is to retain an emphasis on developments of practical use in the construction of modern operating systems. Questions regarding a topic's suitability are welcome and should be sent via electronic mail to the program chair.

Symposium Topics

Topics of interest include (but are not restricted to):

- OS structure and organization
- Microkernel-based systems
- Mach internals and applications
- Distributed systems
- Multiprocessor and MPP systems
- OS support for parallel computing
- Communication paradigms
- Mobile computing
- OS support for real time and multimedia
- OS support for high availability
- OS interaction with HW architecturc
- OS performance analysis and techniques
- OS support for embedded systems
- Security and Privacy

Symposium Overview

The first Symposium will offer two and one half days of technical sessions with presentations of papers selected by the Program Committee. The technical sessions will be preceded by two days of tutorial program, in which several full-day tutorials will be presented by respected instructors. A work-in-progress session will be sponsored; this will be described in later announcements. Papers presented in the technical sessions will be published in the *Proceedings*, which are provided free to technical session attendees and available for purchase from The USENIX Association. Selected paper of particular merit, or possibly the *Proceedings*, are likely also to be distributed to ACM SIGOPS and IEEE TCOS members.

Refereed Papers – How To Submit

Authors must submit an extended abstract by June 21, 1994. The extended abstract should be 5–7 pages in length (excluding references and figures) or about 2500–3500 words. Longer abstracts will be penalized in the review process. The full papers resulting from accepted submissions will go through an editorial review cycle with a member of the program committee, and should end up about 10–14 pages long. Very similar papers must not have been published or submitted for publication elsewhere. Papers accompanied

by so called "non-disclosure agreement" forms are not acceptable and will be returned to the author(s) unread. Note that all submissions are held in the highest confidentiality prior to publication in the *Proceedings*, both as, a matter of policy and in accord with the U.S. Copyright Act of 1976 (Title 17, U.S. Code, Section 102).

The object of an extended abstract is to convince the reviewers that a good paper and 25-minute presentation will result. It is important to identify what has been accomplished, to explain why it is significant, and to compare with prior work in the field, demonstrating knowledge of the relevant literature. The extended abstract should represent the paper in "short form." It must include the abstract as it will appear in the final paper. The body of the extended abstract should be complete paragraphs, not just an outline of the paper. (Sections present in the full paper but omitted from the abstract may be summarized in terse form.) Authors should include full references, figures when available, and, as is often appropriate, performance data. The abstract will be judged on significance, originality, clarity, relevance, and correctness.

For administrative reasons (not blind reviewing), every submission should include one additional page containing:

- Paper title and names and affiliations of authors;
- The name, surface mail address, daytime and evening telephone numbers, e-mail address and (if available) fax number of one of the authors, who will act as the contact to the program committee;
- An indication of which, if any, of the authors are full time students;
- A list of audio/visual equipment desired beyond a microphone and an overhead projector.

Potential authors are strongly encouraged to contact USENIX by e-mail to *osdi-info@usenix.org* or by telephone to +1-510-528-8649 to receive additional guidance about the submission and reviewing process. Those with specific questions are welcome to e-mail to the program chair at *lepreau@cs.utah.edu*.

Where To Submit

Please submit one copy of an extended abstract to the program chair via one of the following methods. All submissions will be acknowledged.

- E-mail Postscript or ASCII to: *osdi-papers@usenix.org*
- Paper mail to: Jay Lepreau, Department of Computer Science, 3190 M.E.B., University of Utah, Salt Lake City, UT USA 84112
- Fax to: +1-801-581-5843 (least preferred method).

Registration Materials

Materials containing all details of the technical sessions and tutorial programs, registration fees and forms, and hotel discount and reservation information will be mailed beginning August 1994. If you wish to receive the registration materials, please contact:

- USENIX Conference Office
  22672 Lambert Street, Suite 613
  Lake Forest, CA USA 92630
  +1-714-588-8649
  Fax: +1-714-588-9706
  Internet: *conference@usenix.org*

Program Committee

- **Program Chair:**
  Jay Lepreau,
  University of Utah

Brian Bershad,
University of Washington

David Black,
OSF Research Institute

Paul Leach,
Microsoft Corporation

Jim Lipkis,
Chorus Systèmes

Karin Petersen,
Xerox PARC

Larry Peterson,
University of Arizona

Karsten Schwan,
Georgia Institute of Techology

Michael Scott,
University of Rochester

Willy Zwaenepoel,
Rice University

Publications from John Wiley & Sons, Inc.

Member Number: _____

All USENIX members receive a 15% discount on the following Wiley publications

Introduction to Client/Server Systems
Paul Renaud
1-57774-X $34.95 ***member price: $29.71***
# of Copies: _____

Portable UNIX
Douglas Topham
1-57926-2 $14.95 ***member price: $12.71***
# of Copies: _____

The UNIX Command Reference Guide
Kaare Christian
1-85580-4 $32.95 ***member price: $28.01***
# of Copies: _____

C++ For Programmers
Leendert Ammeraal
1-93011-3 $32.95 ***member price: $28.01***
# of Copies: _____

Obfuscated C and Other Mysteries
Don Libes
1-57805-3 $39.95 ***member price: $33.96***
# of Copies: _____

Adventures in UNIX Network Applications Programming
Bill Rieken
1-52858-7 $34.95 ***member price: $29.71***
# of Copies: _____

UNIX, Self Teaching Guide
George Leach
1-57924-6 $19.95 ***member price: $16.96***
# of Copies: _____

UNIX Shell Programming, Second Edition
Lowell Jay Arhur
1-51821-2 $29.95 ***member price: $25.46***
# of Copies: _____

Object Oriented Programming withTurbo C++
Keith Weiskamp
1-52466-2 $24.95 ***member price: $21.21***
# of Copies: _____

Berkeley UNIX
A Simple & Comprehensive Guide
James Wilson
1-61582-X $37.95 ***member price: $32.26***
# of Copies: _____

U.S. orders only

☐ Payment enclosed, plus sales tax
☐ VISA ☐ Mastercard
☐ American Express
Card No. _____

Name _____
Firm _____
Address _____
City/State/Zip _____
Signature _____
(order invalid unless signed)

Please mail or fax orders to the following address:

John Wiley & Sons, Inc.
605 Third Avenue
New York, NY 10158
Attn: Karen Cooper

Phone: (212) 850-6789
FAX: (212) 850-6142

WILEY

PTR Prentice Hall is pleased to recommend the following titles to USENIX members...

USENIX members receive a 15% discount on orders

____**Object-Oriented Programming,** Peter Coad and Jill Nicola, 0-13-032616-X
(03261-5) *List:* $47.00 *Members:* $39.95

____**Internetworking with TCP/IP, Vol. III Client Server Programming and Applications for the BSD Socket Version,** Douglas E. Comer and David L. Stevens, 0-13-474222-2
(47422-1) *List:* $51.00 *Members:* $43.35

____**Internetworking with TCP/IP, Vol. III Client Server Programming and Applications for the AT&T TLI Version,** Douglas E. Comer and David L. Stevens, 0-13-474230-3
(47423-9) *List:* $50.00 *Members:* $42.50

____**The Internet Message: Closing the Book with Electronic Mail,** Marshall T. Rose, 0-13-092941-7
(09294-0) *List:* $46.00 *Members:* $39.10

____**Object-Oriented Databases: An Introduction,** Dimitris N. Chorafas and Heinrich Steinmann, 0-13-491804-5
(49180-3) *List:* $36.00 *Members:* $30.60

____**The Anatomy of Programming Languages** Alice E. Fischer and Frances S. Grodzinsky, 0-13-035155-5
(03515-4) *List:* $55.00 *Members:* $46.75

____**The Standard C Library,** PJ Plauger, 0-13-131509-9
(13150-8) *List:* $35.00 *Members:* $29.75

____**All About Administering the NIS+** Rick Ramsey, 0-13-068800-2
(06880-9) *List:* $35.00 *Members:* $29.75

____**The Simple Book: An Introduction to Internet Management** Marshall T. Rose, 0-13-177254-6
(17725-3) *List:* $55.00 *Members:* $46.75

____**Fiber Optics Networks,** Paul Green, Jr., 0-13-319492-2
(31949-1) *List:* $62.00 *Members:* $52.70

____**Networking Operations on UNIX SVR4,** Mike Padavano, 0-13-613555-2
(61355-4) *List:* $44.00 *Members:* $37.40

____**The User's Directory of Computer Networks,** Tracy L. LaQuey, 0-13-950262-9
(95026-1) *List:* $36.00 *Members:* $30.60

____**MIT Project Athena,** George A. Champine, 0-13-585324-9
(58532-3) *List:* $29.00 *Members:* $24.65

____**The Matrix,** John S. Quarterman, 0-13-565607-9
(56560-6) *List:* $50.00 *Members:* $42.50

____**Solaris Porting Guide,** Sunsoft ISV Engineering 0-13-030396-8
(03039-5) *List:* $49.00 *Members:* $41.65

____**Multiprocessor System Architectures,** Catanzaro 0-13-089137-1
(08913-6) *List:* $42.00 *Members:* $35.70

____**The HP-UX System Administrator's "How To" Book** Marty Poniatowski, 0-13-099821-4
(09982-0) *List:* $32.00 *Members:* $27.20

____**UNIX System V Performance Management,** Edited by Phyllis Eve Bregman and Sally A. Browning 0-13-016429-1
(01642-8) *List:* $29.95 *Members:* $25.45

____**SCO® UNIX® Operating System System Administrator's Guide,** Santa Cruz Operation, 0-13-012568-7
(01256-7) *List:* $38.00 *Members:* $32.30

MAIL, PHONE OR FAX YOUR ORDER!

Mention this code when calling: D582-91529-LF(2)

____ Check enclosed, made payable to: ***PTR Prentice Hall***

____ Please bill my: ____VISA ____ MasterCard ____AMEX

Card No. _____

Exp. Date _____ Signature _____

SHIP TO:

Name _____

Company _____

Address _____

_____

City _____ State _____ Zip _____

In USA:
PTR Div.—Prentice Hall Publishing
Box 11073
Des Moines, IA 50381-1073
515-284-6751 FAX 515-284-2607

In Canada:
Prentice Hall Canada
1870 Birchmount Rd.
Scarborough, ON M1P 2J7
416-293-3621 FAX 416-299-2540

In UK, Europe, Middle East & Africa:
Simon & Schuster Int'l
Campus 400, Marylands Ave.
Hemel Hempstead, Hertz HP2 7EZ
442-88 2163 FAX 442-88 2265

FOR MORE INFORMATION, OR QUANTITY ORDERS, PLEASE CALL 201-592-2657

NETWORKS

USENIX members receive a 15% discount on the following MIT Press publications:

GLOBAL NETWORKS
Computers and International Communication

edited by Linda M. Harasim

Global Networks takes up the host of issues raised by the new networking technology that now links individuals, groups, and organizations in different countries and on different continents. The twenty-one contributions focus on the implementation, application, and impact of computer-mediated communication in a global context.

340 pp $29.95 hardcover HARNH

THE NETWORK NATION
Human Communication via Computer
Revised Edition

Starr Roxanne Hiltz and Murray Turoff

"*The Network Nation*... contained a fascinating vision. ...It is a place where thoughts are exchanged easily and democratically and intellect affords one more personal power than a pleasing appearance does. Minorities and women compete on equal terms with white males, and the elderly and handicapped are released from the confines of their infirmities to skim the electronic terrain as swiftly as anyone else." — Teresa Carpenter, *Village Voice*

580 pp $24.95 paperback HILWP

THE EVOLUTION OF C++
Language Design in the Marketplace of Ideas

edited by Jim Waldo

This collection of articles traces the history of C++, from its infancy in the Santa Fe workshop, to its proliferation today as the most popular object-oriented language for microcomputers. Waldo notes in his postscript that in the process of evolving, the language has lost a clearly articulated, generally accepted design center, with no common agreement about what it should or should not do in the future.

279 pp $24.95 paperback WALEP

TECHNOLOGY 2001
The Future of Computing and Communications

edited by Derek Leebaert

Researchers, executives, and strategic planners from inside the companies and laboratories that have shaped today's information age forecast the merging technologies that could well define the computing and communications environment that lies ahead

392 pp $14.95 paperback LEEEP

THE DIGITAL WORD
Text-Based Computing in the Humanities

edited by George P. Landow and Paul Delany

This book explores the larger realm of the knowledge infrastructure where texts are received, reconstructed, and sent over global networks

Technical Communication and Information Systems series 384 pp $39.95 hardcover LANDH

SOCIOMEDIA
Multimedia, Hypermedia, and the Social Construction of Knowledge

edited by Edward Barrett

Sociomedia continues the assessment of hypertext and hypermedia systems begun in *Text, ConText, and HyperText* and *The Society of Text*. It examines the use of integrated multimedia to support social or collaborative research, learning, and instruction in the university, one of the best environments for developing and analyzing the effects of computing technologies on our understanding of complex sets of information

Technical Communications and Information Series 360 pp $50.00 hardcover BARRH

CONNECTIONS
New Ways of Working in the Networked Organization

Lee Sproull and Sara Kiesler

"...Sproull and Kiesler raise crucial questions about our technical and particularly our human strategies as a producing society."
— Howard Webber, *Sloan Management Review*

228 pp $21.95 paperback SPRCP

TECHNOBABBLE

John A. Barry

"A serious study of the language of the new technocracy"
— William Safire, *The New York Times Magazine*

288 pp $12.50 paperback BARCP

Please send me these titles _____

☐ Payment enclosed ☐ Purchase Order Attached Charge to my: ☐ Master Card ☐ VISA

Card # _____ exp. _____ Signature _____

$ _____ Total for book(s)
$ 3.00 Postage for North American addresses
$ _____ Canadian customers add 7% GST
$ _____ Total for book(s) & postage

Name _____
Address _____
City _____ State _____ Zip _____
Phone _____ FAX _____

Make checks payable and send order to:

THE MIT PRESS
55 Hayward Street, Cambridge, MA 02142-1399 USA

To order by phone, call (617) 625-8569 or (800) 356-0343. E-mail order # mitpress-orders@mit.edu The operator will need this code **UNIX1.**

McGraw–Hill Books
20% Discount for USENIX Members

Qty

Open Systems Interconnection Its Architecture and Protocols, Revised Edition.....

Bijendra N. Jain & Ashok K. Agrawala 0-07-032385-2 hardcover
retail: $50.00 member price: $40.00

A Students Guide to UNIX.....

Harley Hahn 0-07-025511-3 paperback
retail: $25.75 member price: $20.60

Data Network Design Packet Switching, Frame Relay, 802.6/DQDB.....

Darren L. Spohn 0-07-060360-X hardcover
retail: $59.00 member price: $47.20

Migrating to Open Systems: Taming the Tiger.....

Daniel R. Perley, 0-07-707778-4 hardcover
retail: $30.00 member price: 24.00

I am a member of the USENIX Association. Please send me the books I have indicated at the member special rate. I have added $3.00 postage and handling for the first book ordered, $1.00 for each additional book, plus my local sales tax.

Total price of books _____
Sales tax _____
Postage & Handling _____
Total enclosed _____

USENIX Membership #_____

____ Check or money order is enclosed - payable to McGraw-Hill, Inc.

____ Charge my ____ Visa ____ Mastercard ____ Discover ____ Amex

Account #_____

Expiration Date_____

Signature_____

Shipping Information (7/93 - 03US001)

Name _____

Address _____

_____

Daytime Phone # _____

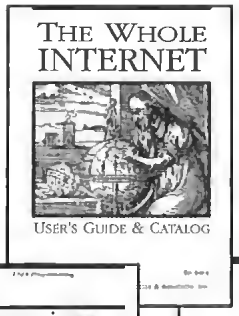
Send orders to:

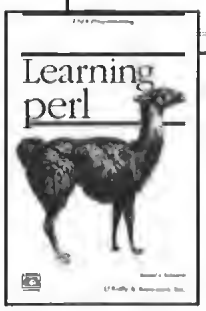
McGraw-Hill, Inc.
Attn: Rosa Perez
11 West 19th Street - 4th Floor
New York, New York 10011

If I am not completely satisfied, I will return the book(s) within 15 days or a full refund or credit.
Satisfaction unconditionally guaranteed. Prices subject to change without notice. We can only accept orders within the continental USA.

FROM O'REILLY & ASSOCIATES, INC.

10% OFF TO USENIX MEMBERS
Mention "Usenix" & your membership number when ordering.





Books that help people get more out of computers

The Whole Internet User's Guide & Catalog

By Ed Krol, 1st Edition September 1992
400 pages, ISBN 1-56592-025-2

A comprehensive—and best-selling—introduction to the Internet, the international network that includes virtually every major computer site in the world.
This book is a comprehensive introduction to what's available and how to find it. In addition to electronic mail, file transfer, remote login, and network news, *The Whole Internet* pays special attention to some new tools for helping you find information. Whether you're a researcher, a student, or just someone who likes electronic mail, this book will help you to explore what's possible.

Learning perl

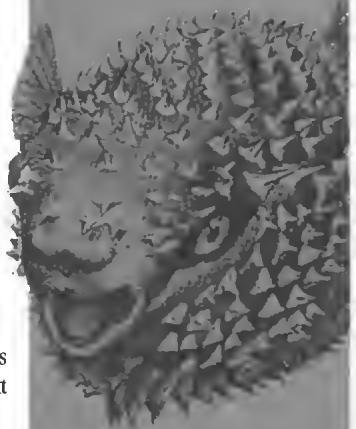
By Randal L. Schwartz
1st Edition November 1993
274 pages, ISBN 1-56592-042-2

Perl is rapidly becoming the "universal scripting language." Combining capabilities of the UNIX shell, the C programming language, *sed*, *awk*, and various other utilities, it has proved its use for tasks ranging from system administration to text processing and distributed computing. *Learning perl* is a step-by-step, hands-on tutorial designed to get you writing useful perl scripts as quickly as possible. In addition to countless code examples, there are numerous programming exercises, with full answers. For a comprehensive and detailed guide to programming with Perl, read O'Reilly's companion book *Programming perl*.

Understanding Japanese Information Processing

By Ken Lunde
1st Edition September 1993
470 pages, ISBN 1-56592-043-0

Understanding Japanese Information Processing provides detailed information on all aspects of handling Japanese text on computer systems. It covers everything from the origins of modern-day Japanese to the latest information on specific emerging computer encoding standards.



sendmail

By Bryan Costales, with Eric Allman & Neil Rickert
1st Edition November 1993
830 pages, ISBN 1-56592-056-2

Although *sendmail* is used on almost every UNIX system, it's one of the last great uncharted territories—and most difficult utilities to learn—in UNIX system administration. This book provides a complete *sendmail* tutorial, plus extensive reference material. It covers the BSD, UIUC IDA, and V8 versions of *sendmail*. Part One of the book is a tutorial on understanding *sendmail*; Part Two covers practical issues in *sendmail* administration; Part Three is a comprehensive reference section; and Part Four consists of appendices and a bibliography.

ORACLE Performance Tuning

By Peter Corrigan & Mark Gurry
1st Edition September 1993
642 pages, ISBN 1-56592-048-1

The ORACLE relational database management system is the most popular database system in use today. This book shows you the many things you can do to dramatically increase the performance of your existing ORACLE system, whether you are running RDBMS Version 6 or Version 7. You may find that this book can save you the cost of a new machine; at the very least, it will save you a lot of headaches.

Migrating to Fortran 90

By James F. Kerrigan
1st Edition November 1993 (est.)
315 pages (est.), ISBN 1-56592-049-X

This book is a practical guide to Fortran 90 for the current Fortran programmer. It provides a complete overview of the new features that Fortran 90 has brought to the Fortran standard, with examples and suggestions for use. Topics include array sections, modules, allocatable arrays and pointers, file handling, and numeric precision.

O'Reilly & Associates, Inc.

103A Morris Street, Sebastopol, California 95472 • 800/998-9938 • 707/829-0515 • FAX 707/829-0104 • order@ora.com

Open Systems is Your Destination. UniForum Gets You There First.

Discover New Products, New Technologies and New Ideas to help you face the challenges of an Open Systems future.

Over 400 leading companies will make UniForum '94 one of the richest IS resources for Open Systems professionals anywhere, anytime. Thousands of essential products and services that can help you achieve your productivity and profitability goals will be available for you to evaluate.

Special Exhibit Events and Demonstrations include:

- UniForum World of the Future Pavilion
- DCE Demonstration
- New Product Award Competition
- Access to the Data Highway
- Virtual Reality
- And more

All New 5 Day Conference Program*

Designed to meet the needs of both the UNIX and Open Systems Veteran and the IS Professional from proprietary environments, UniForum's conference program has the answers.

Monday & Tuesday, March 21-22 Special 2-day programs

- Second Annual Technology Managers' Conference From Pilot to Production: Implementing Client/Server Applications in The Real World
- USENIX/SAGE Systems Administration Tutorials For Systems Managers and Administrators

Tutorials - Intensive One Day Sessions

- Performance Measurement/Management and Capacity Planning of UNIX Systems
- Basics of Object-Oriented Technology
- TCP/IP: Understanding the Protocols of the Internet
- Enterprise Messaging
- Distributed Systems Management
- Internet For Beginners
- Multimedia Systems: A Guided Tour
- Internetwork Management In Transition: Moving From SNMP Version 1 to SNMP Version 2
- Introduction to ATM
- Mastering Fundamental UNIX

*As of 10/7/93 Subject to Change

Plenary Session - FREE to All Attendees

Wednesday, March 23
Politics of Open Architectures: Sincerely Expanding Open Systems or Manipulating the Market for Acceptance

Thursday, March 24
Windows NT: UNIX Threat or Paper Tiger?

Friday, March 25
Comparison of Hardware Vendors Open Systems Strategies

Workshops - 1/2 Day Hands-On Classroom Sessions

Friday, March 25

- Exchanging E-mail Between Two Systems
- Synchronizing E-mail Directories Between or Among Two or More Systems
- Exchanging Files Between Two Systems
- Properly Executing an Application Running on Another System

Seven Conference Tracks Cover The Issues

Wednesday thru Friday, March 23-25

- Global Networks And The Role of UNIX
- A New Age For Enterprise System Strategies
- Open Systems:Truths and Myths
- Costs And Benefits – A Manager's Perspective On UNIX
- UNIX And Large Scale Data Management
- Trends In Software Development
- UniForum Opens The Future

For more information call toll free (800) 225-4698 (in U.S.) or (508) 879-6700. For fastest service, fill in the coupon below and fax to: (508)872-8237.

UNIFORUM 1994
MARCH 21-25 • MOSCONE CENTER • SAN FRANCISCO
Your Roadmap to an Open Future

Fax to (508) 872-8237 or mail to: UniForum '94 c/o IDG, World Expo, P.O. Box 9107, Framingham, MA 01701- 9107

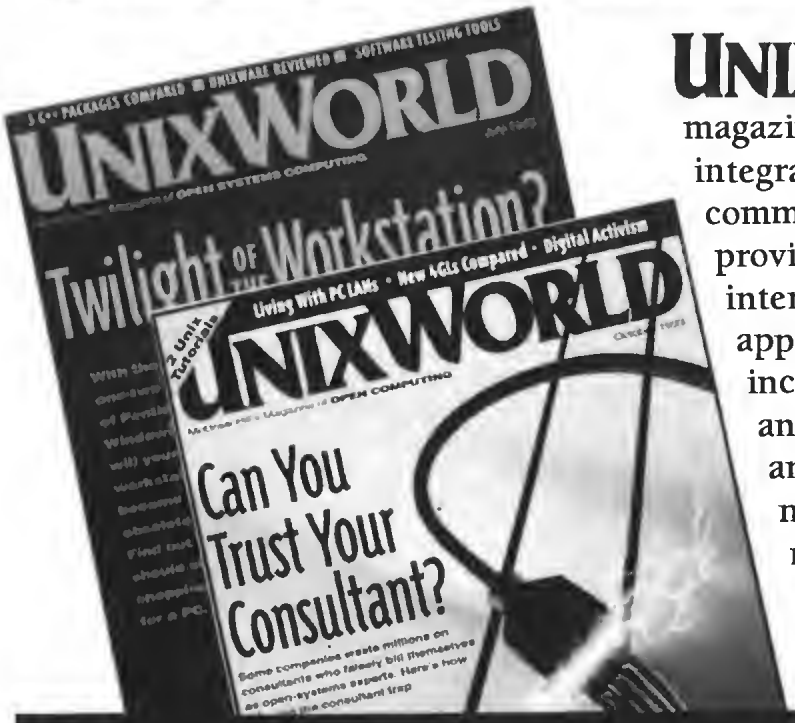
Name _____
Title _____
Company _____
Address _____
City _____ State _____
Zip _____
Code USE

UniForum'94 is sponsored by UniForum, the International Association of Open Systems Professionals.

Special Offer

USENIX Association Members

THE WORLDWIDE COMPUTING AUTHORITY



UNIXWORLD is a monthly magazine written for people who integrate, resell, manage, and program commercial computer systems to provide solutions based on interoperable networks and applications. Topics for features include the effective use of UNIX and other open systems to downsize and distribute applications. The magazine also provides a useful mix of product reviews, case studies, industry profiles and analysis, and technical tutorials.

THE BENEFITS OF UNIXWORLD:

Programming Tips - helps the reader become educated in UNIX and OPEN SYSTEMS computing. Provides real-world application information the reader can use everyday.

Unbiased Hardware and Software Reviews - helps the reader recommend and justify purchasing decisions.

Clear Editorial - so both beginners and experts can get full benefit from the publication. Product Trends and Technical Advances - to aid professionals in developing company-wide systems strategies.

Profiles on U.S. and International Companies - provides the reader with a truly global perspective of the marketplace and the players. Helps vendors assess and monitor the competition.

Special Rates:

DOMESTIC		CANADIAN		OTHER INTERNATIONAL (Air Delivery)	
1 Yr.	$15.00	1 Yr.	$21.00	1 Yr.	$ 48.00
2 Yrs.	$24.00	2 Yrs.	$36.00	2 Yrs.	$ 90.00
3 Yrs.	$30.00	3 Yrs.	$48.00	3 Yrs.	$130.00

Be wise subscribe today!

To place your order please call:

Kelly Kendall at 1-800-327-6850 or fax your order to 609/426-7055

McGraw-Hill, Inc.• Dept. 418 • Hightstown, NJ 08520

Publications Order Form

Conference & Workshop Proceedings

Qty	Proceedings		Member Price	Non-Member Price	Subtotal	Overseas Postage	Total
	WINTER/SUMMER CONFERENCES						
___	San Francisco	Winter '94	30	39	$_____	$20	$_____
___	Cincinnati	Summer '93	25	33	$_____	18	$_____
___	San Diego	Winter '93	33	40	$_____	25	$_____
___	San Antonio	Summer '92	23	30	$_____	14	$_____
___	San Francisco	Winter '92	30	39	$_____	22	$_____
___	Nashville	Summer '91	32	38	$_____	22	$_____
___	Dallas	Winter '91	28	34	$_____	18	$_____
___	Anaheim	Summer '90	22	22	$_____	15	$_____
___	Washington, DC	Winter '90	25	25	$_____	15	$_____
___	Baltimore	Summer '89	20	20	$_____	15	$_____
___	San Diego	Winter '89	30	30	$_____	20	$_____
___	San Francisco	Summer '88	29	29	$_____	20	$_____
___	Dallas	Winter '88	26	26	$_____	15	$_____
___	Phoenix	Summer '87	35	35	$_____	20	$_____
___	Washington, DC	Winter '87	10	10	$_____	15	$_____
___	Atlanta	Summer '86	37	37	$_____	20	$_____
___	Denver	Winter '86	25	25	$_____	15	$_____
___	Portland	Summer '85	45	45	$_____	25	$_____
___	Dallas	Winter '85	15	15	$_____	10	$_____
___	Salt Lake City	Summer '84	29	29	$_____	20	$_____
___	Washington, DC	Winter '84	25	25	$_____	15	$_____
___	Toronto	Summer '83	32	32	$_____	20	$_____
___	San Diego	Winter '83	28	28	$_____	15	$_____
	LARGE INSTALLATION SYSTEMS ADMINISTRATION						
___	LISA VII	Nov. '93	25	33	$_____	12	$_____
___	LISA VI	Oct. '92	23	30	$_____	12	$_____
___	LISA V	Sept. '91	20	23	$_____	11	$_____
___	LISA IV	Oct. '90	15	18	$_____	8	$_____
___	LISA III	Sept. '89	13	13	$_____	9	$_____
___	LISA II	Nov. '88	8	8	$_____	5	$_____
___	LISA I	April '87	4	4	$_____	5	$_____
	C++						
___	C++ Conference	Aug. '92	30	39	$_____	20	$_____
___	C++ Conference	April '91	22	26	$_____	11	$_____
___	C++ Conference	April '90	28	28	$_____	18	$_____
___	C++ Conference	Oct. '88	30	30	$_____	20	$_____
___	C++ Workshop	Nov. '87	30	30	$_____	20	$_____
	SECURITY						
___	UNIX Security IV	Oct. '93	15	20	$_____	9	$_____
___	UNIX Security III	Sept. '92	30	39	$_____	11	$_____
___	UNIX Security II	Aug. '90	13	16	$_____	8	$_____
___	UNIX Security	Aug. '88	7	7	$_____	5	$_____
	MACH						
___	Mach Symposium III	April '93	30	39	$_____	18	$_____
___	Mach Symposium	Nov. '91	24	28	$_____	14	$_____
___	Mach Workshop	Oct. '90	17	20	$_____	9	$_____

Qty	Proceedings		Member Price	Non-Member Price	Subtotal	Overseas Postage	Total
DISTRIBUTED & MULTIPROCESSOR SYSTEMS (SEDMS)							
___	SEDMS IV	Sept. '93	24	32	$_____	14	$_____
___	SEDMS III	Mar. '92	30	36	$_____	20	$_____
___	SEDMS II	Mar. '91	30	36	$_____	20	$_____
___	SEDMS	Oct. '89	30	30	$_____	20	$_____
MICROKERNELS & OTHER KERNEL ARCH.							
___	Microkernels & Other Kernel... II	Sept. '93	15	20	$_____	9	$_____
___	Microkernels & Other Kernel... I	Apr. '92	30	39	$_____	20	$_____
GRAPHICS							
___	Graphics Workshop V	Nov. '89	18	18	$_____	10	$_____
___	Graphics IV	Oct. '87	10	10	$_____	10	$_____
___	Graphics III	Nov. '86	10	10	$_____	5	$_____
___	Graphics II	Dec. '85	7	7	$_____	5	$_____
OTHER WORKSHOPS							
___	Mobile Computing	Aug. '93	15	20	$_____	8	$_____
___	File Systems	May '92	15	20	$_____	9	$_____
___	UNIX Transaction Processing	May '89	12	12	$_____	8	$_____
___	Software Management	Apr. '89	20	20	$_____	15	$_____
___	UNIX & Supercomputers	Sept. '88	20	20	$_____	10	$_____

Discounts are available for bulk orders. Please inquire.

Total price of Proceedings _____ **
Calif. residents add sales tax _____
Total overseas postage _____
Total enclosed _____

****If you are paying member price, please include member's name and/or membership number**_____

PAYMENT OPTIONS*

___ Check enclosed - payable to USENIX Association

___ Charge my: ___ VISA ___ MC Account # _____ Exp.Date _____

___ Purchase order enclosed Signature _____

* Outside the USA? Please make your payment in US currency by one of the following:
- Check - issued by a local branch of a US Bank
- Charge (VISA, MasterCard, or foreign equivalent)
- International postal money order

Shipping Information

Please allow 2 – 3 weeks for delivery. Overseas orders are shipped via air printed matter.

Please mail or fax this order form with payment to:

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
FAX 510/548-5738

Ship to: _____

- If you are not a member and wish to receive our membership information packet, please check this box. ☐

Local User Groups

The Association will support local user groups by doing a mailing to assist in the formation of a new group and publishing information on local groups in *;login:*. At least one member of the group must be a current member of the Association. Send additions and corrections to: <*login@usenix.org*>.

CA – Fresno:

The Central California UNIX Users Group consists of a uucp-based electronic mailing list to which members may post questions or information. For connection information:

Educational and governmental institutions:
Brent Auernheimer (209) 278-2573,
<*brent@CSUFresno.edu* or *csufres!brent*>

Commercial institutions or individuals:
Gordon Crumal (209) 251-2648
<*csufres!gordon*>

CA – Orange County:

Meets the 2nd Monday of each month

UNIX Users Association of Southern California
Paul Muldoon (714) 556-1220 ext. 137
New Horizons Computer Learning Center
1231 E. Dyer Rd., Suite 140
Santa Ana, CA 92705

CO – Boulder:

Meets monthly at different sites. For meeting schedule, send email to <*fruug-info@fruug.org*>.

Front Range UNIX Users Group
Software Design & Analysis, Inc.
1113 Spruce St., Ste. 500
Boulder, CO 80302
Steve Gaede (303) 444-9100
<*gaede@fruug.org*>

D.C.– Washington, D.C.:

Meets 1st Tuesday of each month.

Washington Area UNIX Users Group
9811 Mallard Drive
Laurel, MD 20708
Alan Fedder (301) 953-3626

FL – Coral Springs

S. Shaw McQuinn (305) 344-8686
8557 W. Sample Road
Coral Springs, FL 33065

FL - Melbourne:

Meets the 3rd Monday of every month.

Space Coast UNIX User's Group
Steve Lindsey (407) 242-4766
<*lindsey@vnet.ibm.com*>

FL – Orlando:

Meets the 3rd Thursday of each month.

Central Florida UNIX Users Group
Mikel Manitius (407) 444-8448
<*mikel@aaa.com*>

FL – Western:

Meets 1st Thursday of each month.

Florida West Coast UNIX Users Group
Richard Martino (813) 536-1776
Tony Becker (813) 799-1836
<*ucrsys!tony*>
Ed Gallizzi, Ph.D. (813) 864-8272
<*e.gallizzi@compmail.com*>
Jay Ts (813) 979-9169
<*uunet!pdn!tscs!metran!jan*>
Dave Lewis (407)242-4372
<*dhl@ccd.harris.com*>

GA – Atlanta:

Meets on the 1st Monday of each month in White Hall, Emory University.

Atlanta UNIX Users Group
P.O. Box 12241
Atlanta, GA 30355-2241
Mark Landry (404) 365-8108

KS or MO – Kansas:

Meets on 2nd Monday of each month.

Kansas City UNIX Users Group (KUUG)
813B Street
Blue Springs, MO 64015
(816) 235-5212
<*mlg@cstp.umkc.edu*>

MI – Detroit/Ann Arbor

Meets on the 2nd Thursday of each month in Ann Arbor.

Southeastern Michigan Sun Local Users Group and Nameless UNIX Users Group
Steve Simmons office: (313)769-4086
home: (313) 426-8981
<*scs@lokkur.dexter.mi.us*>

MN – Minneapolis/St. Paul:

Meets the 1st Wednesday of each month.

UNIX Users of Minnesota
17130 Jordan Court
Lakeville, MN 55044
Robert A. Monio (612) 220-2427
<*pnessutt@dmshq.mn.org*>

MO – St. Louis:

St. Louis UNIX Users Group
P.O. Box 2182
St. Louis, MO 63158
Terry Linhardt (314) 772-4762
<*uunet!jgaltstl!terry*>

NE– Omaha:

Meets monthly.

/usr/group/nebraska
P.O. Box 31012
Omaha, NE 68132
Phillip Allendorfer (402) 423-1400

New England – Northern:

Meets monthly at different sites.

Peter Schmitt 603) 646-2085
Kiewit Computation Center
Dartmouth College
Hanover, NH 03755
<*peter.schmitt@dartmouth.edu*>

NJ – Princeton:

Meets monthly.

Princeton UNIX Users Group
Mercer County Community College
1200 Old Trenton Road
Trenton, NJ 08690
Peter J. Holsberg (609) 586-4800
<*mccc!pjh*>

NM – Albuquerque:

ASIGUNIX meets every 3rd Wednesday of each month. Phil Hortz 505/275-0466.

NY – New York City:

Meets every other month in Manhattan.

Unigroup of New York City
G.P.O. Box 1931
New York, NY 10116
<*unigroup@murphy.com*>
Bob Young
(212) 490-8470

OK - Tulsa:

Meets 2nd Wednesday of each month.

Tulsa UNIX Users Group, $USR
Stan Mason (918) 560-5329
<*tulsix!smason@drd.com*>
Mark Lawrence (918) 743-3013
<*mark@drd.com*>

TX – Austin:

Meets 3rd Thursday of each month.

Capital Area Central Texas UNIX Society
P.O. Box 9786
Austin, TX 78766-9786
<*officers@cactus.org*>
Tom Painter (512) 835-5457
<*president@cactus.org*>

TX – Dallas/Fort Worth:

Meets the 1st Thursday of each month.

Dallas/Fort Worth UNIX Users Group
P.O. Box 867405
Plano, TX 75086
Evan Brown (214) 519-3577
<*evbrown@dsccc.com*>

TX – Houston:

Meets 3rd Tuesday of each month.

Houston UNIX Users Group
(Hounix) answering machine (713) 684-6590
Bob Marcum, President (713) 270-8124
Chuck Bentley, Vice-president
(713) 789-8928
<*chuckb@hounix.uucp*>

WA – Seattle:

Meets monthly.

Seattle UNIX Group Membership Info.
Bill Campbell (206) 947-5591
6641 East Mercer
Mercer Island, WA 98040-0820
<*bill@celestial.com*>

CANADA – Toronto:

143 Baronwood Court
Brampton, Ont. Canada L6V 3H8
Evan Leibovitch (416) 452-0504
<*evan@telly.on.ca*>

CANADA – Ottawa:

The Ottawa Carleton UNIX Users Group
D.J. Blackwood (613)957-9305
<*dave@revcan.rct.ca*>

LISA Groups

Back Bay LISA (BBLISA)

New England forum covering all aspects of system and network administration, for large and small installations. Meets monthly, at MIT in Cambridge, MA.

For information, contact:

J.R. Oldroyd
Open Advisors Limited
(617) 227-5635
<jr@opal.com>

Mailing list subscription:
<requests: bblisa-request@cs.umb.edu>
Mailing list postings: *<bblisa@cs.umb.edu>*

For current calendar of events:
<finger bblisa@cs.umb.edu>

BAY LISA

The Bay-LISA group meets monthly in Santa Clara, CA to discuss topics of interest for administration of sites with more than 100 users and/or computers.

Send email to *<baylisa-info@sysadmin.com>*
or contact:

Bjorn Satdeva
408/241-3111
<bjorn@sysadmin.com>

Prime Time Freeware

by Rich Morin
<rdm@cfcl.com>

Prime Time Freeware for Perl, Volume 1, Number 1

PTF-Perl 1-1 contains a definitive collection of Perliana: documentation, scripts, source code, and pre-built binaries. The associated book contains the entire Perl manual page and reference card, augmented by useful introductory material and indexes. The issue consists of one CD-ROM and a 150+ page introductory book.

Prime Time Freeware for UNIX, Volume 3 Number 1

PTF 3-1 contains 1200 MB of compressed (GNU Zip) archives. 5000 MB of current, interesting source code and documentation, organized and annotated for your hacking pleasure. The issue consists of two CD-ROMs and a 100+ page introductory book.

Prime Time SDK for UNIXWare, Volume 2, Number 1

2-1 builds on our earlier work, but adds several new features. It is a complete plug and play Software Development Kit, with compilers, editors, formatters, GUI builders, windowing tools, and more. The issue includes XFree86-2.0, a greatly enhanced and optimized version of X. We are also pleased to announce the inclusion of CDT, a freeware-based integrated programming environment. The issue consists of one CD-ROM and a 100+ page introductory book.

Ordering and Prices

PTF issues normally cost $60, but SUG and USENIX members may purchase them for $50. California residents must add 7% tax. US shipping/handling charges are $4/order + $1/unit. Foreign S/H is $6/order + $4/unit.

PTF accepts MasterCard and Visa, postal money orders in US funds, and checks in US funds that are payable through a US bank. Orders and inquiries should be directed to:

Prime Time Freeware
370 Altair Way, #150
Sunnyvale, CA 94086

TEL: 408/433-9662
FAX: 408/433-0727
Email: *<ptf@cfcl.com>*

Calendar of Events

1994

Mar 14-16 Interex ICMS, New Orleans, LA
21-25 * UniForum, San Francisco, CA
Apr 4 - 8 * SANS III, Arlington, VA
11-14 * C++, Cambridge, MA
11-15 EurOpen Forum, Egham, Surrey, UK
18 -22 IEEE 1003, Lake Tahoe, CA
25-28 * UNIX Applications Development Toronto, Canada
May 2- 6 NetWorld+INTEROP 94, Las Vegas, NV
7-13 DECUS, New Orleans, LA
Jun 6-10 * USENIX, Boston, MA
" NetWorld+INTEROP 94, Berlin
16-18 SUG, San Francisco, CA
Jul 11-15 IEEE 1003
Aug 1- 2 * High–Speed Networking, Berkeley, CA
Sept 6 - 9 AUUG, Melbourne, Australia
12-14 NetWorld+INTEROP 94, Atlanta, GA
18 -22 Interex 94, Denver, CO
20 -22 GUUG, Wiesbaden, Germany
19 -23 * LISA VIII, San Diego, CA
Oct 17-21 IEEE 1003
23-27 ACM OOPSLA, Portland, OR
26-28 * Very High Level Languages, Santa Fe, NM
Nov 12-18 DECUS, Anaheim, CA
14-18 * OSDI, Monterey, CA
SUG Technical Workshop, Austin, TX

1995

Jan 16-20 * USENIX, New Orleans, LA
Feb 21-23 UniForum, Dallas, TX
May 13-19 DECUS, Atlanta, GA
Jun 19-23 * USENIX, San Francisco, CA
Aug 13-17 Interex 95, Toronto, Canada
Nov 2- 8 DECUS, San Francisco, CA

1996

Jan 22-26 * USENIX, San Diego, CA
Mar 12-14 UniForum, San Francisco, CA
May 18-24 DECUS, Orlando, FL
June 17-21* Washington, DC
Aug 4 - 8 Interex 96, San Diego, CA
Nov 16-22 DECUS, Anaheim, CA

This is a combined calendar of planned conferences, symposia, and standards meetings related to the UNIX operating system. If you have a UNIX-related event that you wish to publicize, please contact <*login@usenix.org*>. Please provide your information in the same format as above.

* = events sponsored by the USENIX Association.

ACM: Association for Computing Machinery
AUUG: Australian UNIX Users Group
DECUS: Digital Equipment Computer Users Society
EurOpen: European Forum for Open Systems

FedUNIX: Council of Advanced Computing Systems Technologists in Government
GURU: Roumanian UNIX User Group
GUUG: German UNIX Users Groups

IEEE: Institute of Electrical and Electronics Engineers
IETF: Internet Engineering Task Force
INET: Internet Society
Interex: Intl. Association of Hewlett-Packard Comp.Users

JUS: Japan UNIX Society
LISA: USENIX Systems Administration Conference
NOSSDAV: Network and Operating System Support for Digital Audio and Video

OOPSLA: Object - oriented Programming Systems, Languages, and Applications

OSDI: Symposium on Operating Systems Design & Implementation
SAGE: System Administrators' Guild
SANS: System Administration, Networking & Security

SUG: Sun User Group
UKUUG: United Kingdom UNIX Systems Users Group
UniForum: International Association of UNIX and Open Systems Professionals

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

Second Class Postage
PAID
At Berkeley, California
and Additional Offices

POSTMASTER: Send address changes to *;login:*, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710

UPCOMING SYMPOSIA AND CONFERENCES

MARCH 21–22, 1994

USENIX/SAGE SECURITY AND SYSTEMS ADMINISTRATION MANAGEMENT SEMINARS
at the UniForum Conference

Moscone Convention Center,
San Francisco, California

APRIL 4–8, 1994

1994 WORLD CONFERENCE ON TOOLS AND TECHNIQUES FOR SYSTEM ADMINISTRATION, NETWORKING, AND SECURITY (SANS-III)

Stouffers Concourse Hotel
Arlington, Virginia

Sponsored by FedUNIX, the USENIX Association and SAGE, the System Administrators Guild, and others

APRIL 11–14, 1994

6TH C++ CONFERENCE & ADVANCED TOPICS WORKSHOP

Marriott Hotel
Cambridge, Massachusetts

Program Chair: Doug Lea
State University of New York at Oswego

APRIL 25–28, 1994

UNIX APPLICATIONS DEVELOPMENT SYMPOSIUM

Marriott Hotel
Toronto, Ontario, Canada

Program Chair: Jim Duncan
Pennsylvania State University

Program Vice Chair: Greg Woods
GAW Consulting

JUNE 6–10, 1994

SUMMER 1994 TECHNICAL CONFERENCE

Marriott Hotel
Boston, Massachusetts

Program Chairs: Margo Seltzer
Harvard University, and Keith Bostic
University of California, Berkeley

AUGUST 1–2, 1994

SYMPOSIUM ON HIGH-SPEED NETWORKING
with field trips on August 3

Claremont Hotel & Resort
Oakland, California

Program Chair: Pat Parseghian,
AT&T Bell Laboratories

TO RECEIVE FULL INFORMATION

Please contact: USENIX Conference Office, 22672 Lambert St., Suite 613, Lake Forest, CA USA 92630
+1 (714) 588-8649; FAX: +1 (714) 588-9706; e-mail: conference@usenix.org